Practical Electronics

The UK's premier electronics and computing maker magazine

PIC n' Mix
PIC18F development board

Make it with Micromite
Counting pulses and rotary encoders

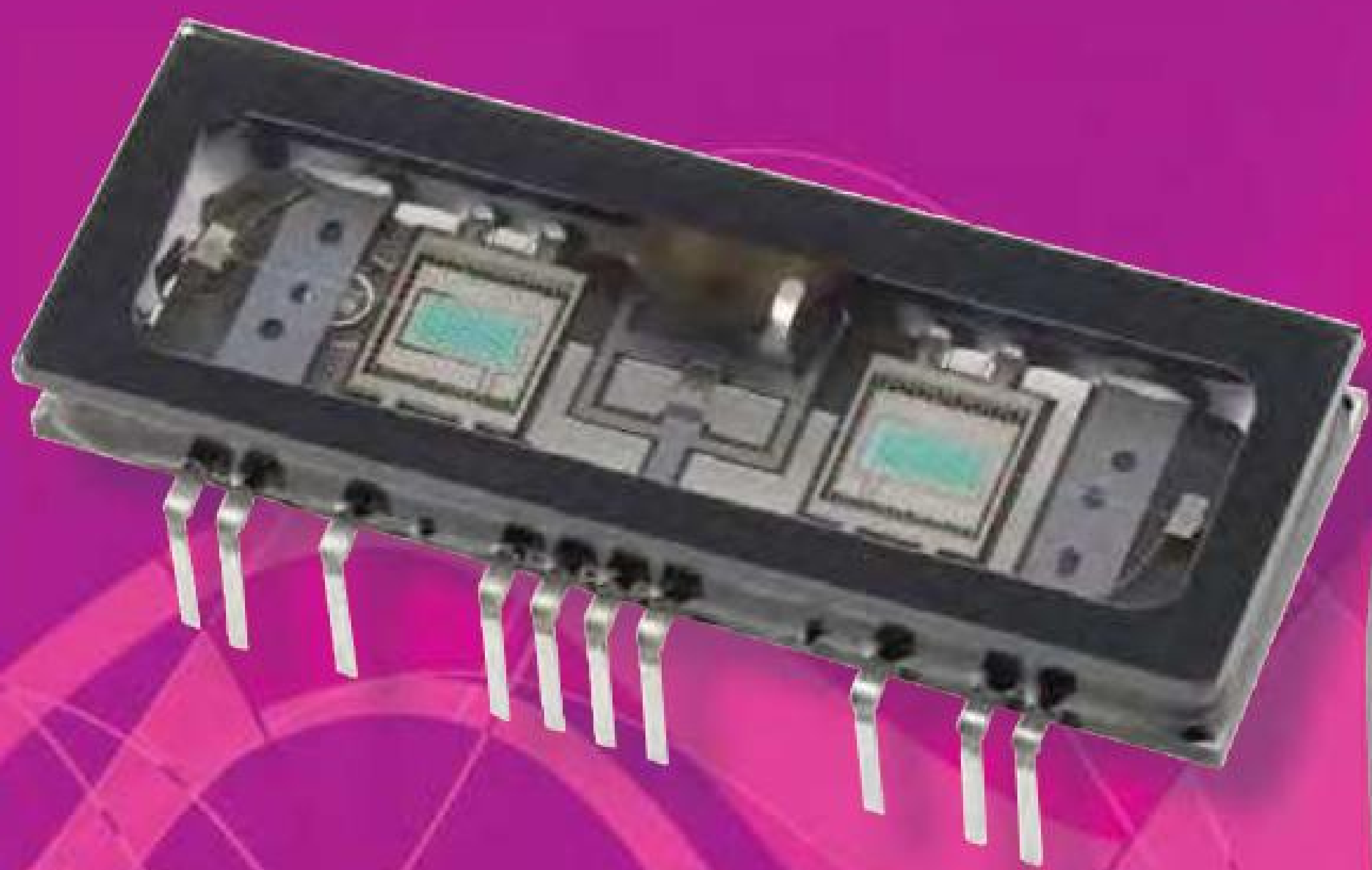
Audio Out
Germanium BJT amplifier

Circuit Surgery
Interference and noise

Toot toot! Complete Arduino DCC Controller

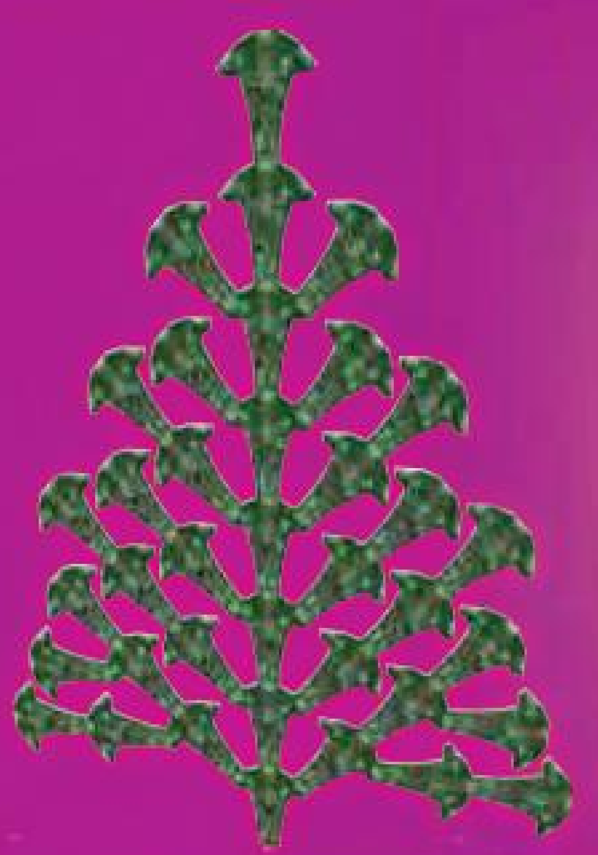
WIN!
Microchip SAM IoT WG Development Board
WIN!

NeoPixel gamma correction

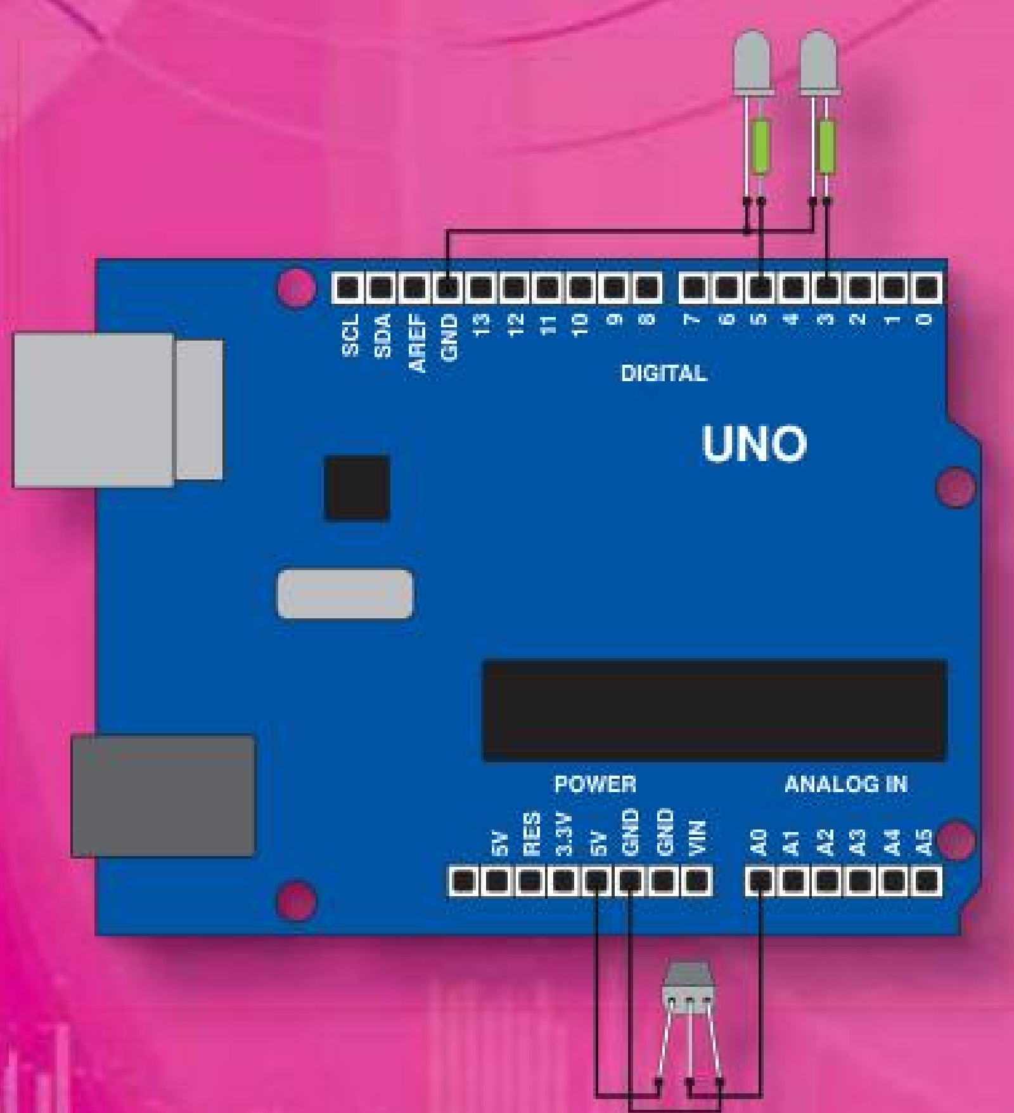


SILICON CHIP Nutube Preamplifier
Volume
Power

Amazing Nutube miniature valve stereo preamplifier



Fun LED Christmas Tree offer!



Fascinating XOD/Arduino thermometer

PLUS!

Techno Talk – Lights out for dangerous bulbs?
KickStart – *NEW* series! – using the 2N7000 MOSFET
Net Work – New electric cars and Android TV

Jan 2021 £4.99

www.electronpublishing.com | @practicalelec | practicalelectronics

MOTION
DETECTED

Compact, Powerful MCUs

PIC18-Q41 Family for Improved Sensor Interface Designs

The PIC18-Q41 family of microcontrollers (MCUs) combines sophisticated analog peripherals and powerful Core Independent Peripherals (CIPs) for small, high-performance data acquisition and sensor-interfacing applications. Available in small 14- and 20- pin packages, these MCUs are equipped with an operational amplifier, a 12-bit Analog-to-Digital Converter with Computation, and 8-bit Digital-to-Analog Converters providing a high level of analog integration for amplification, filtering and signal conditioning. With our comprehensive development tool suite, you can easily configure peripherals and functions, generate application code and simulate analog circuits prior to hardware prototyping to reduce your development time and speed your time to market. PIC18-Q41 MCUs are well suited for IoT edge nodes, medical, wearables, LED lighting, home automation, automotive and industrial process control.

Key Features

- Up to 64 KB Flash program memory/up to 4 KB data SRAM/512B data EEPROM
- Small-footprint packages for sensor interface applications
- Faster time to market with MPLAB® Code Configurator and MPLAB Mindi™ Analog Simulator
- Easy-to-use development tools to get your design to market faster

MICROCHIP
PIC18F16Q41

www.microchip.com/Q41

The Microchip name and logo, the Microchip logo and MPLAB are registered trademarks and Mindi is a trademark of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks are the property of their registered owners.
© 2020 Microchip Technology Inc. All rights reserved. DS40002255A. MEC2345-ENG-11-20

MICROCHIP

Practical Electronics

Volume 50. No. 1
January 2021
ISSN 2632 573X

Contents

Projects and Circuits

Nutube Miniature Valve Stereo Preamplifier by John Clarke 14
Korg and Noritake Itron of Japan recently released their Nutube 6P1 twin triode. It's party piece is that it can operate from just a few.

Complete Arduino DCC Controller by Tim Blythman 26
Build this fantastic Arduino-compatible controller shield that forms the basis of a DCC system. It can also be used as a DCC booster or high-current motor driver.

Intelligent 8x8 RGB LED matrix by Jim Rowe 34
Master a module with an 8x8 matrix of 64 'intelligent' RGB LEDs. The LEDs are controlled via a single wire, and multiple modules can be cascaded.

Series, Features and Columns

The Fox Report by Barry Fox 8
HDMI capture dongle update

Techno Talk by Mark Nelson 10
Bad boy bulbs

Net Work by Alan Winstanley 12
The latest news on electric cars, tips and tricks for running an Android-based TV system and advice on setting up a two-screen Windows PC.

KickStart by Mike Tooley 38
Part 1: MOSFET switching devices in linear applications – introducing the 2N7000

PIC n' Mix by Mike Hibbett 43
Part 3: PIC18F development board

Audio Out by Jake Rothman 46
Theremin Audio Amplifier – Part 3

Make it with Micromite by Phil Boyce 50
Part 24: Counting pulses, rotary encoders and a digital safe

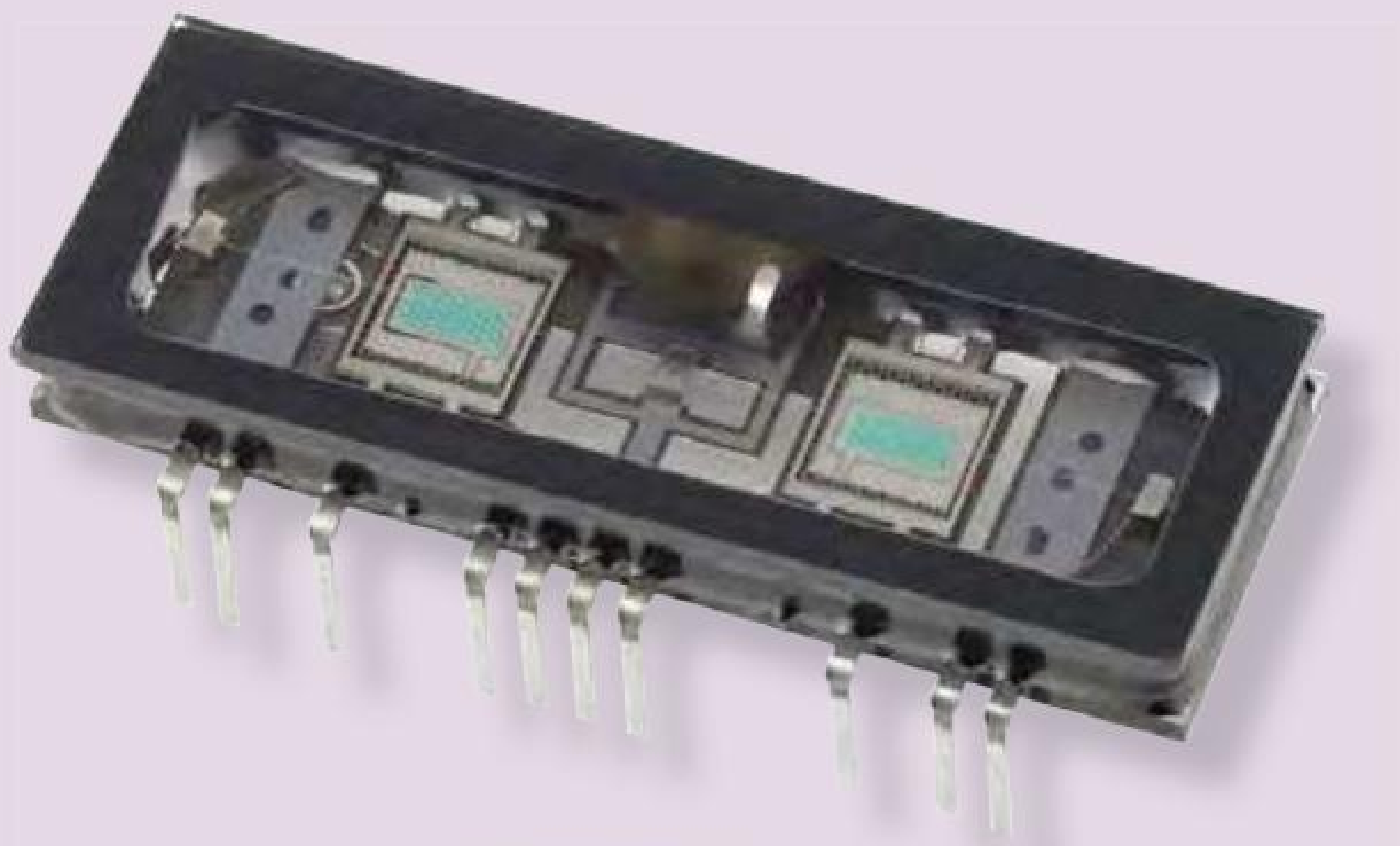
Circuit Surgery by Ian Bell 54
Interference and noise

Max's Cool Beans by Max The Magnificent 58
Flashing LEDs and drooling engineers – Part 11

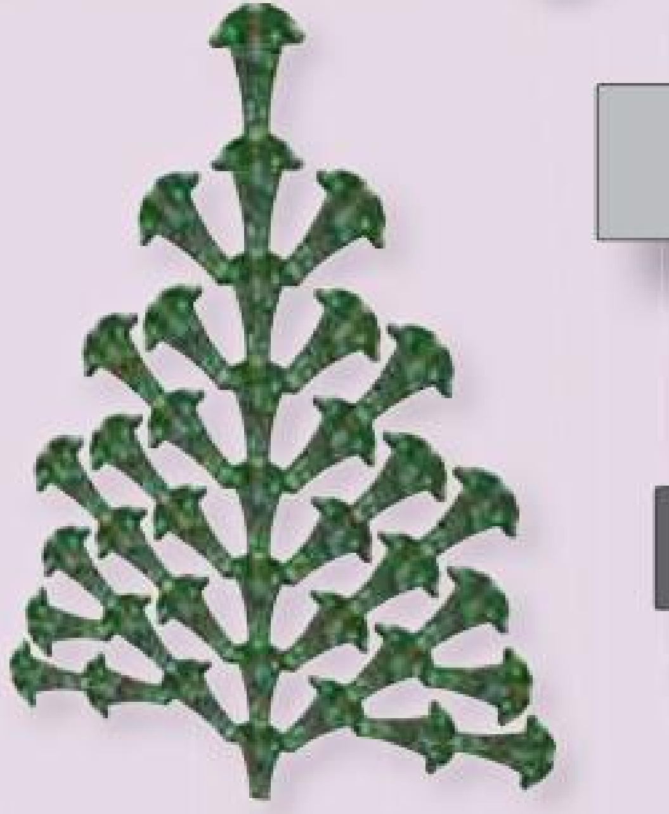
Visual programming with XOD by Julian Edgar 63
Light Column Thermometer

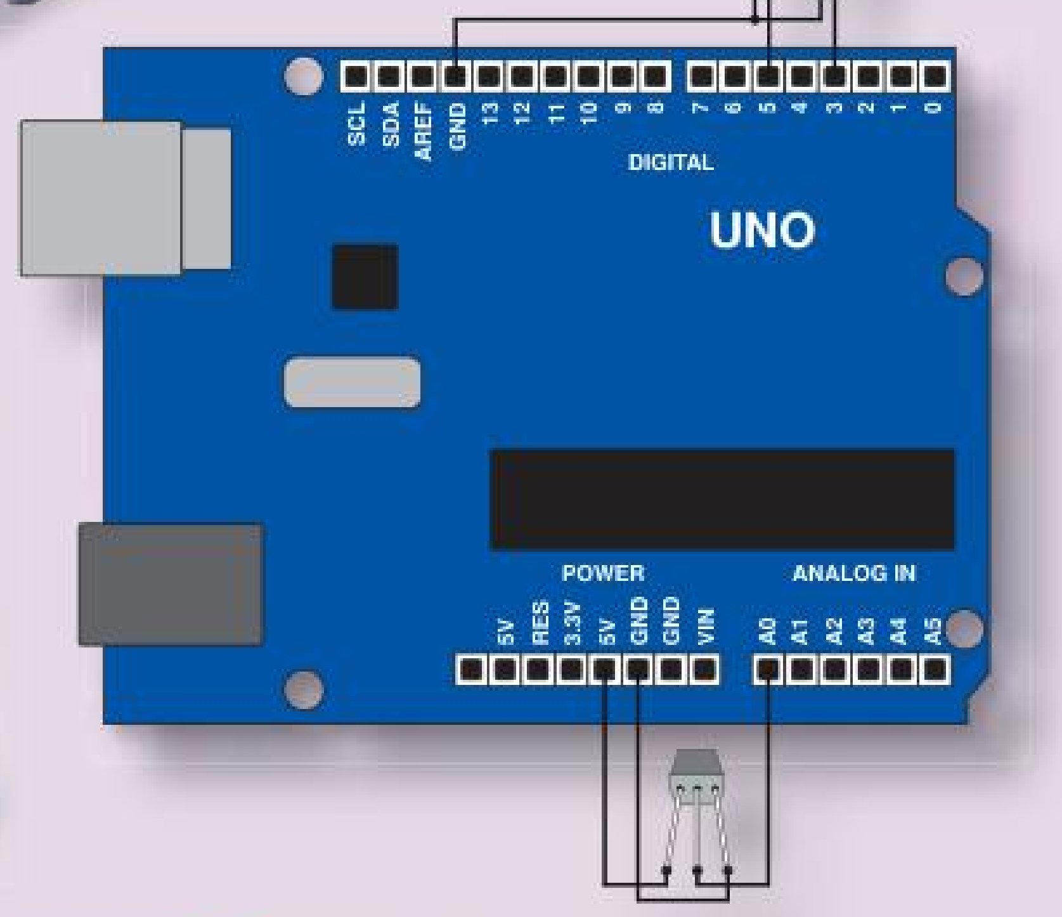
Regulars and Services

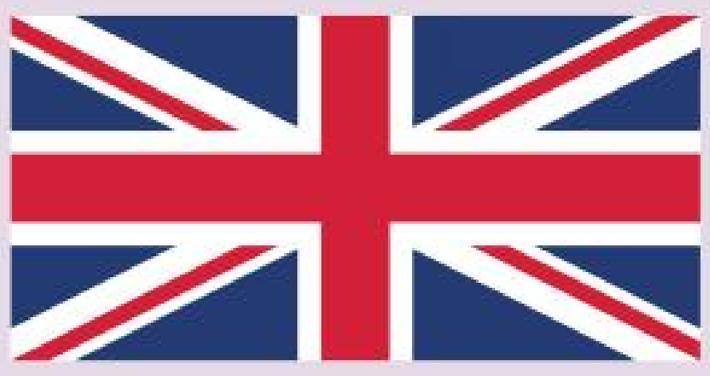
Wireless for the Warrior 2
Subscribe to Practical Electronics and save money 4
NEW! Practical Electronics back issues DOWNLOADS – great 20-year deal! 6
Reader services – Editorial and Advertising Departments 7
Editorial 7
Looking forward to 2021
Exclusive Microchip reader offer 11
Win a Microchip SAM IoT WG Development Board
LED Christmas Tree offer 25
Practical Electronics – get your back issues here! 25
PE Teach-In 9 42
PE Teach-In 8 66
Classified ads and Advertiser index 67
Practical Electronics PCB Service 68
PCBs for Practical Electronics projects
Direct Book Service 70
Build your library of carefully chosen technical books
Next month! – highlights of our next issue of Practical Electronics 72



SILICON CHIP Nutube Preamplifier
Power
Volume







Made in the UK.

Written in Britain, Australia, the US and Ireland.

Read everywhere.

© Electron Publishing Limited 2020

Copyright in all drawings, photographs, articles, technical designs, software and intellectual property published in *Practical Electronics* is fully protected, and reproduction or imitation in whole or in part are expressly forbidden.

The February 2021 issue of *Practical Electronics* will be published on Thursday, 7 January 2021 – see page 72.

WIRELESS FOR THE WARRIOR

by LOUIS MEULSTEE

THE DEFINITIVE TECHNICAL HISTORY OF RADIO COMMUNICATION EQUIPMENT IN THE BRITISH ARMY

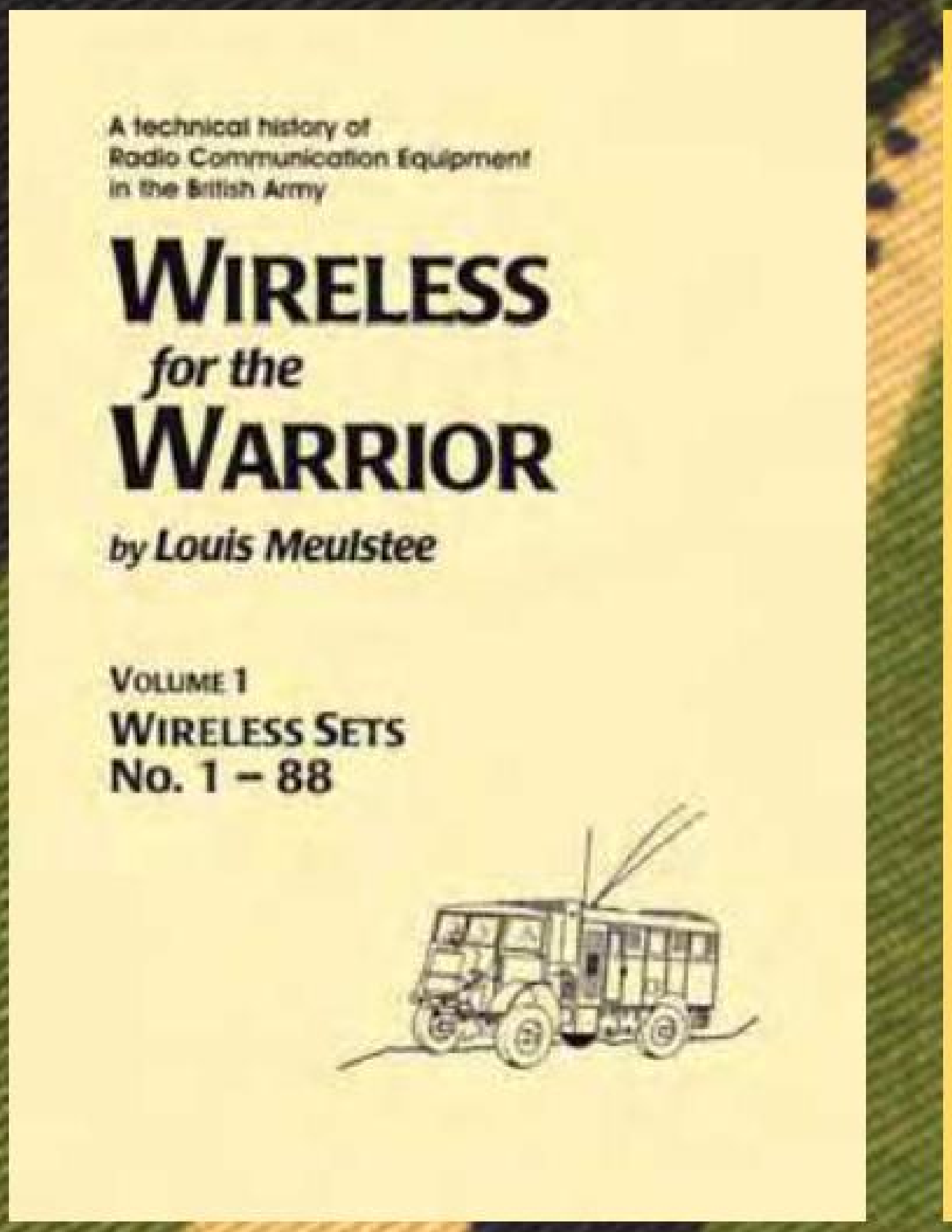
The *Wireless for the Warrior* books are a source of reference for the history and development of radio communication equipment used by the British Army from the very early days of wireless up to the 1960s.

The books are very detailed and include circuit diagrams, technical specifications and alignment data, technical development history, complete station lists and vehicle fitting instructions.

Volume 1 and *Volume 2* cover transmitters and transceivers used between 1932-1948. An era that starts with positive steps taken to formulate and develop a new series of wireless sets that offered great improvements over obsolete World War I pattern equipment. The other end of this timeframe saw the introduction of VHF FM and hermetically sealed equipment.

Volume 3 covers army receivers from 1932 to the late 1960s. The book not only describes receivers specifically designed for the British Army, but also the Royal Navy and RAF. Also covered: special receivers, direction finding receivers, Canadian and Australian Army receivers, commercial receivers adopted by the Army, and Army Welfare broadcast receivers.

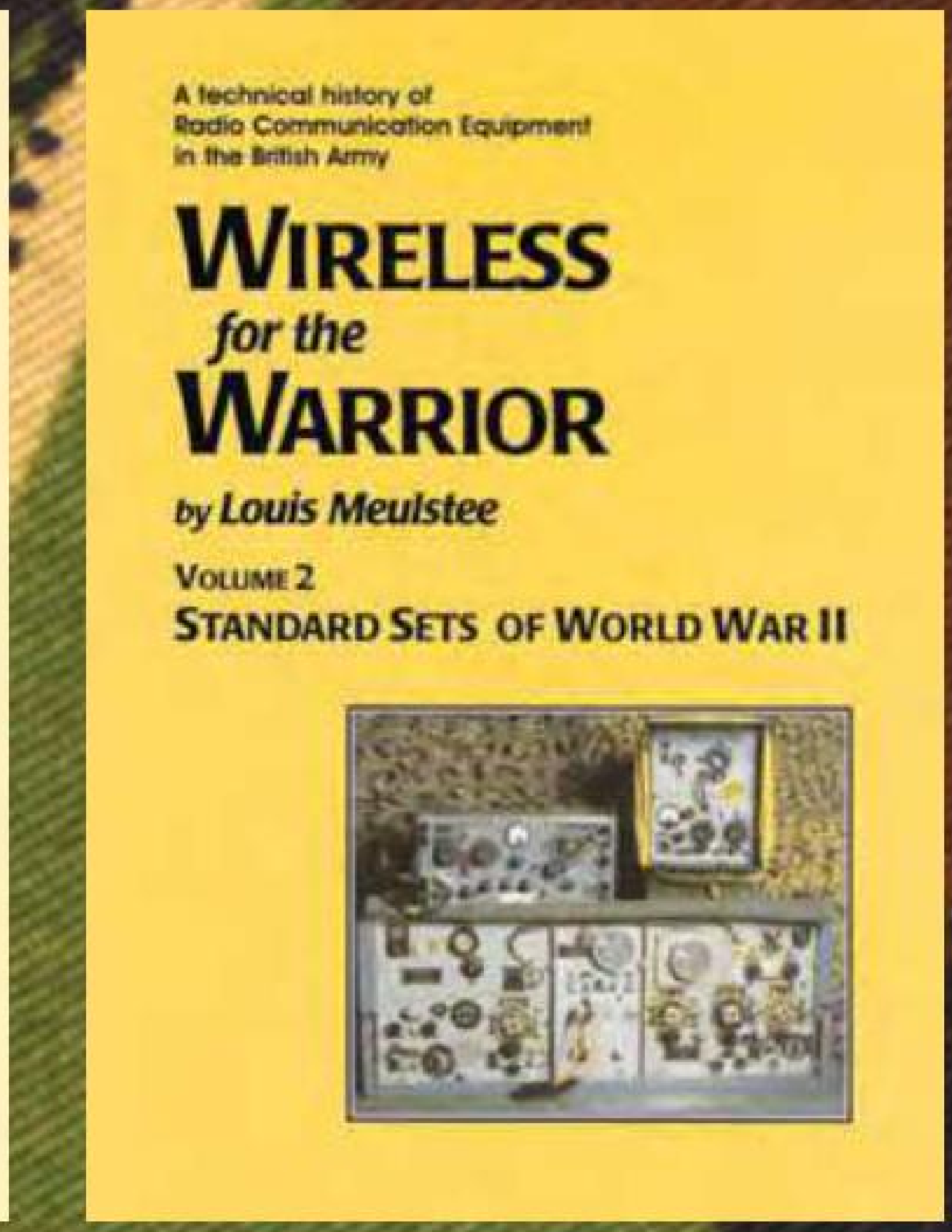
Volume 4 covers clandestine, agent or 'spy' radio equipment, sets which were used by special forces, partisans, resistance, 'stay behind' organisations, Australian Coast Watchers and the diplomatic service. Plus, selected associated power sources, RDF and intercept receivers, bugs and radar beacons.



A technical history of
Radio Communication Equipment
in the British Army

WIRELESS
for the
WARRIOR
by Louis Meulstee

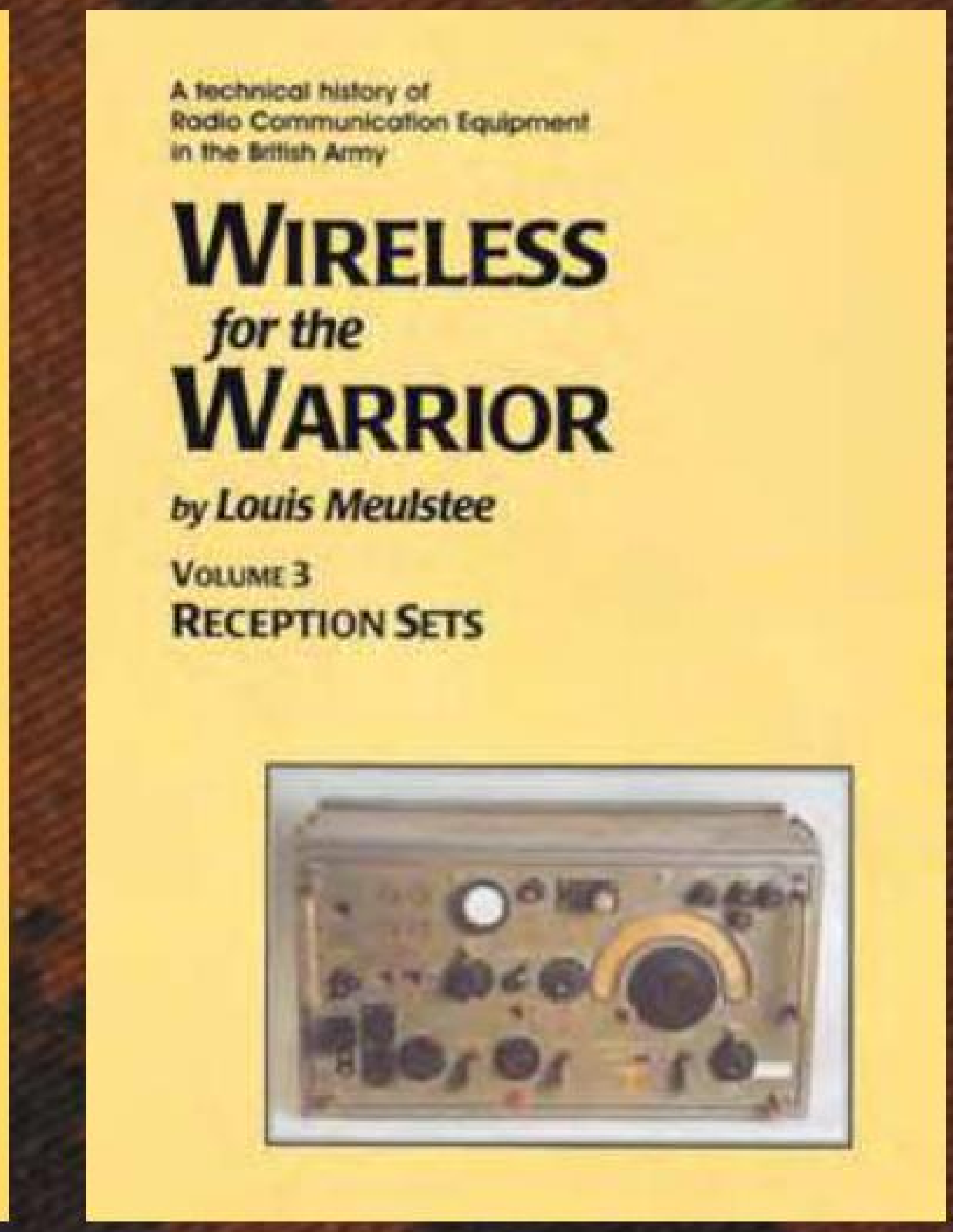
VOLUME 1
WIRELESS SETS
No. 1 – 88



A technical history of
Radio Communication Equipment
in the British Army

WIRELESS
for the
WARRIOR
by Louis Meulstee

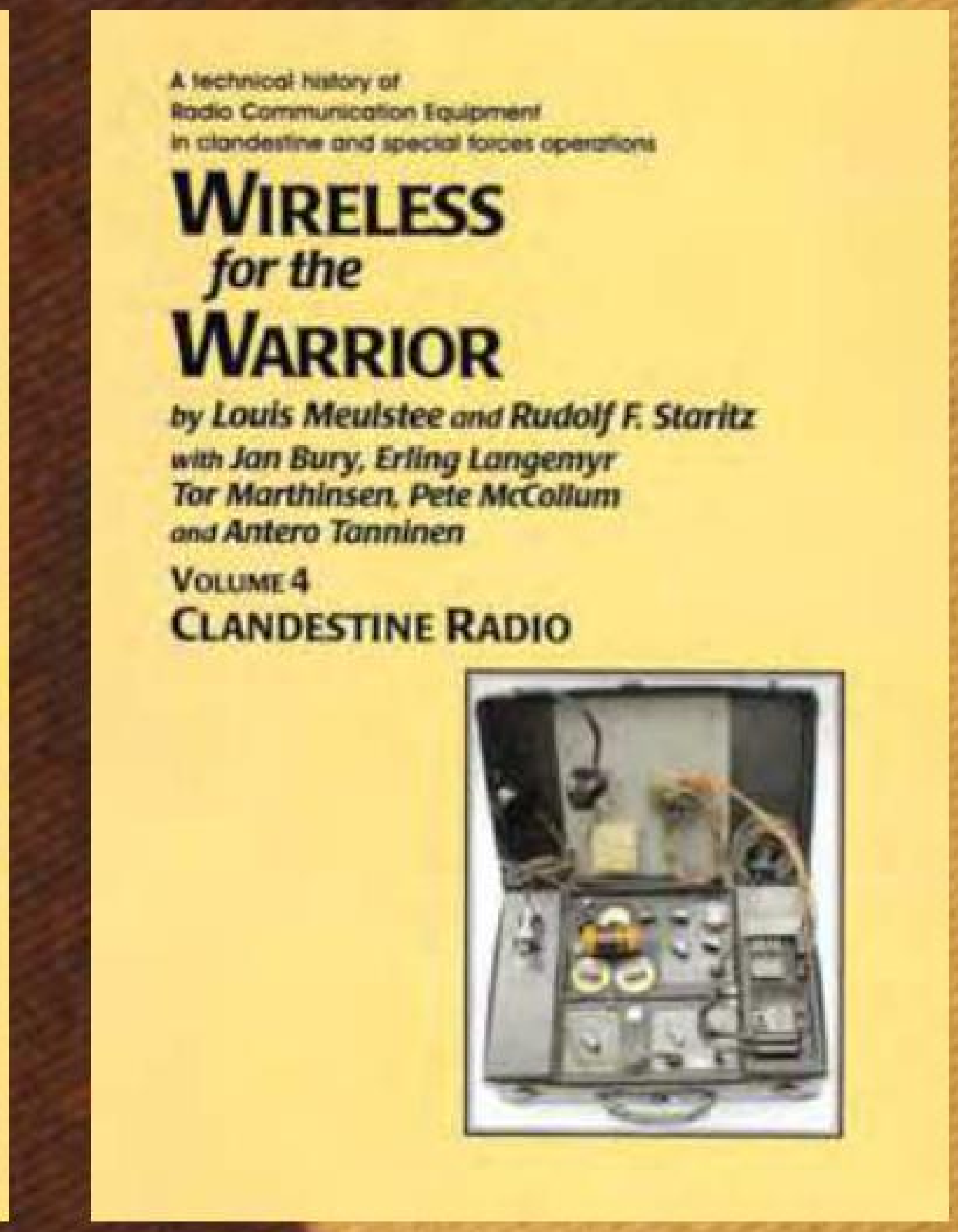
VOLUME 2
STANDARD SETS OF WORLD WAR II



A technical history of
Radio Communication Equipment
in the British Army

WIRELESS
for the
WARRIOR
by Louis Meulstee

VOLUME 3
RECEPTION SETS



A technical history of
Radio Communication Equipment
in clandestine and special forces operations

WIRELESS
for the
WARRIOR
by Louis Meulstee and Rudolf F. Staritz
with Jan Bury, Erling Langemyr
Tor Marthinsen, Pete McCollum
and Antero Tanninen

VOLUME 4
CLANDESTINE RADIO

ORDER YOURS TODAY!
JUST CALL 01202 880299 OR VISIT www.electronpublishing.com

QUASAR
electronics

Quasar Electronics Limited
PO Box 6935, Bishops Stortford
CM23 4WP, United Kingdom

Tel: 01279 467799
E-mail: sales@quasarelectronics.co.uk
Web: quasarelectronics.co.uk

All prices include 20% VAT. Free UK mainland delivery on orders over £60.
Postage & Packing Options (Up to 0.5Kg gross weight): **UK Standard 2-5 Day Delivery** - £4.95 : **UK Mainland Next Day Delivery** - £9.95 : Please order online if you reside outside the UK (our website will calculate postage for you).
Payment: We accept all major credit/debit cards. Make UK cheques/PO's payable to Quasar Electronics Limited and include P&P detailed above.
!! Order online for reduced price postage and fast despatch !!
Please visit our online shop now for full details of over 1000 electronic kits, projects, modules and publications. Discounts for bulk quantities.

QuasarElectronics.co.uk

Solutions for Home, Education & Industry Since 1993

cebek

velleman

Official Main Dealer stocking the full range of Kits, Modules, Robots, Instruments, Tools and much, much more...

SANTA'S Workshop

Electronic Kits & Modules

We have a massive selection of self-assembly electronic kits and pre-assembled modules. Please see the full range on our website or call for details.

LED Buddy / LED Tester Kit

Hold any type of LED to the contact pads to see it's polarity, forward voltage & the recommended series resistor value. Adjustable target current & forward voltage. Great design aid. 1x PP3 battery powered.
Order Code: MK198 - ***£13.92***

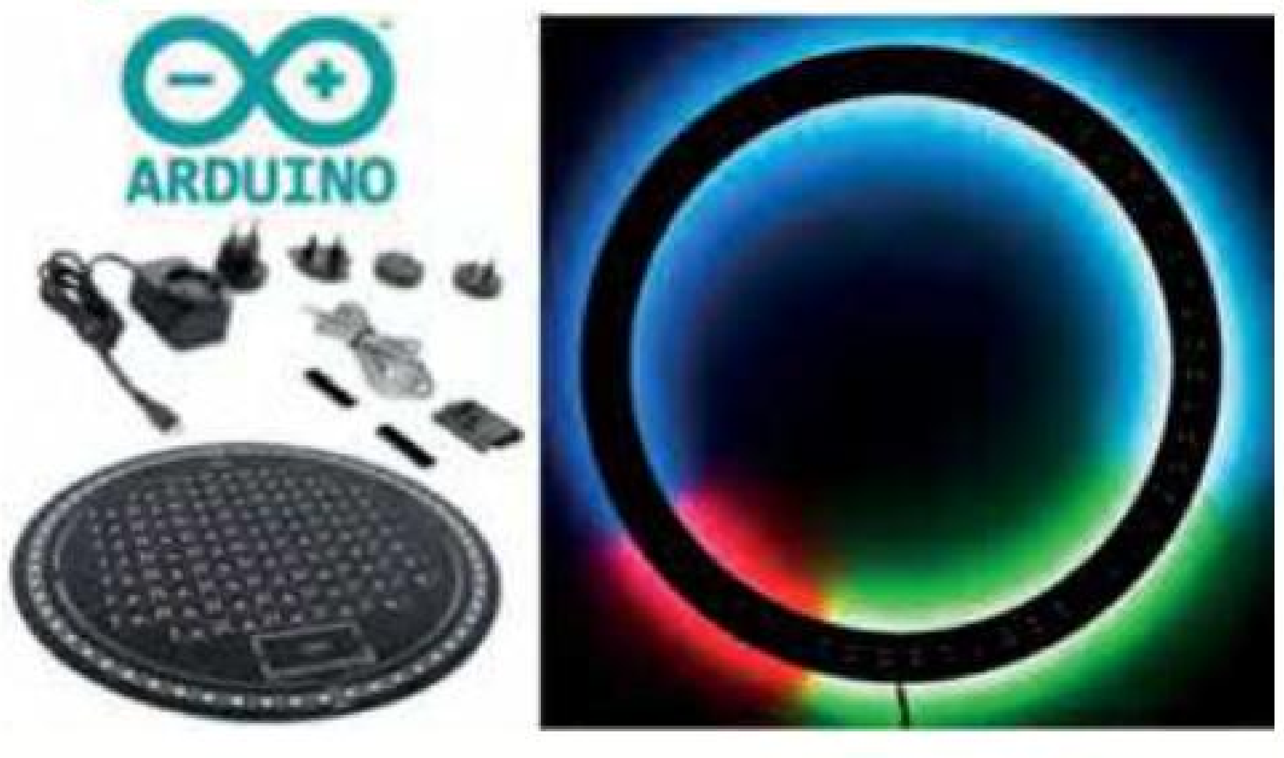
3-in-1 All Terrain Robot Kit

Multi-function tracked mobile robot. Transform into 3 amazing models using different track modules: Forklift, Rover (shown above) and Gripper. The wired controller commands the robot to make it move forward, backward, turn, grip or lift. Solderless assembly. Requires 4x AA batteries.
Order Code: KSR11 - ***£34.40***

Snowman Flashing LED Kit

Have some educational festive fun with this animated LED snowman gadget. 69 multicoloured LEDs (great for soldering practice). Snowflake effect with PWM controlled LEDs. Random generator for a more realistic effect. 1x 9V PP3 battery or 9-12Vdc wall adapter powered. Makes an ideal gift.
Order Code: MK200 - ***£20.34***

Brightdot Clock Kit - BLACK Edition



Brighten any room or space with this fully Arduino® compatible, ESP32 controlled BrightDot clock kit. This designer black edition features 60 bright RGB LEDs that reflect against the surface on which you mount the clock, hence telling you what time of day it is. ESP32 data cable & power supply included.
Order Code: K2400B - ***£117.43***

DIY Electronic Watch Kit

Make your own DIY, Arduino compatible electronic wrist watch! 24 amber coloured LEDs are bright enough to be clearly visible in broad daylight! Pre-programmed with an addictive reflex game and of course with a basic time view. You can easily re-program it to your liking by using open-source Arduino® library and the K1201 Custom Cradle Kit or a USB to UART module (neither included).
Order Code: K1200 - ***£23.94***

Stereo Ultrasonic Bat Detector Kit

Converts high frequency sounds (20 - 90kHz) normally imperceptible to humans like bat signals into audible noise. Can also help detect failures in machines, engines, etc. Stereo feature adds the possibility to pinpoint the source. Requires 3xAA batteries. 3.5mm jack output for headphones.
Order Code: K8118 - ***£21.59***

LED Christmas Tree Kit

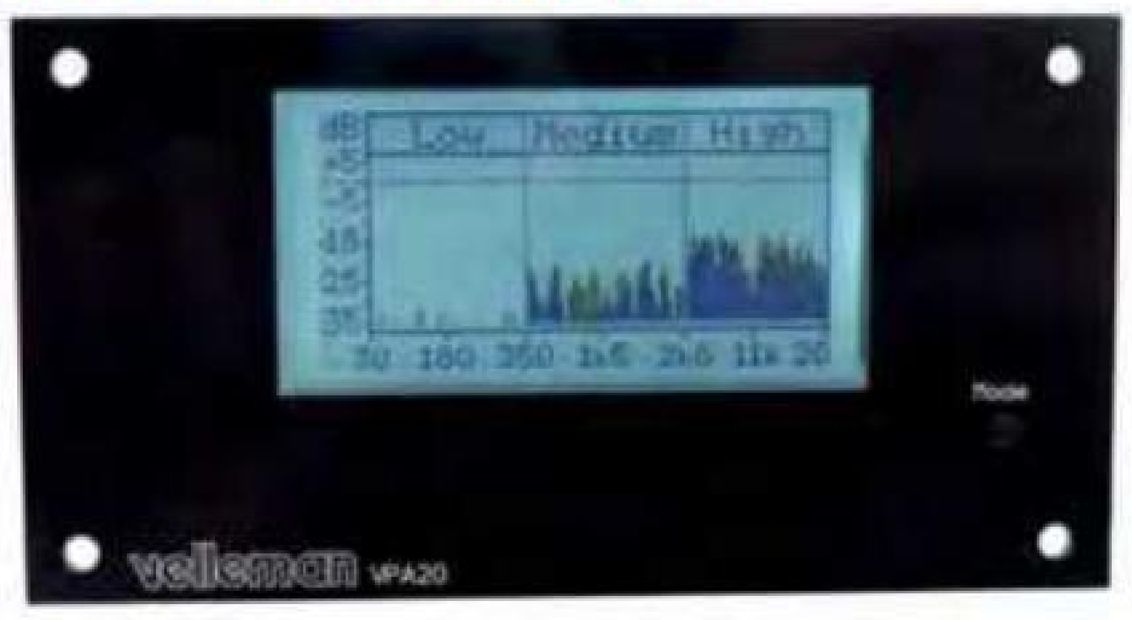
Ho! Ho! Ho! The classic Christmas kit for the budding electronics enthusiast. 15 blinking blue LEDs. Requires 1xPP3 battery. Get cosy…
Order Code: MK100B - ***£7.19***

Card Sales & Enquiries 01279 467799

Digitally Controlled FM Radio Kit

Build your own modern, high quality FM receiver project with excellent sensitivity powered by a simple 9V PP3 battery (not included. Auto-seeking button. 4 station presets. Volume control. Excellent learning project for schools and colleges.
Order Code: MK194N - ***£20.39***

Audio Analyser Display Kit

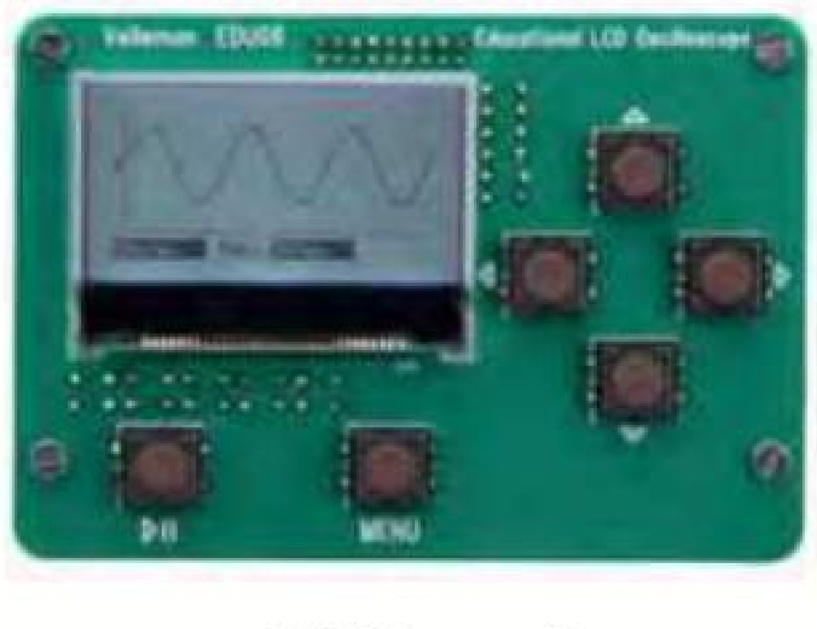


Small, compact LCD display, ideal for panel mounting. Give your homemade audio gear a high-tech look. Upgrade existing equipment. Provides Peak Power, RMS Power, Mean dB, Peak dB, Linear Audio Spectrum And 1/3 Octave Audio Spectrum. Auto / Manual range selection. Peak-hold function. Speaker impedance selection.
Order Code: K8098 - ***£39.54***

Electronic Component Tester Kit

Build your own versatile component tester. Shows value and pin layout information for resistors (0.1 Ohm resolution, max. 50 MOhm), coils (0.01mH - 20H), capacitors (28pF - 100mF), diodes, BJT, JFET, E-IGBT, D-IGBT, E-MOS & D-MOS.
Order Code: K8115 - ***£44.34***

LCD Oscilloscope Educational Kit



Build your own LCD oscilloscope with this exciting new kit. Learn how to read signals. See the electronic signals you learn about displayed on your own LCD oscilloscope. Despite the low cost, this oscilloscope kit has a lot of features found on expensive units like signal markers, frequency, dB, true RMS readouts and more. A powerful auto-setup function will get you going in a flash!
Order Code: EDU08 - ***£48.54***

Practical Electronics

UK readers SAVE £1 on every issue

NEW subscriptions hotline!

01202 087631

Take out a one-year subscription and save £10 over the year.
Even better – save £1 per issue if you subscribe for two years – a total saving of £24.

Overseas rates represent exceptional value

You also:
- **Avoid any cover price increase for the duration of your subscription**
- **Get your magazine delivered to your door each month**
- **Guarantee a copy, even if the newsagents sell out**

Order by:
- **Phone or post with a cheque, postal order or credit card**
- **Or order online at: www.electronpublishing.com**

Practical Electronics

SUBSCRIPTION PRICES

Subscriptions for delivery direct to any address in:

UK: 6-months £26.99, 12-months £49.85, 24-months £94.99

Europe Airmail: 6-months £30.99, 12-months £57.99, 24-months £109.99

Rest Of The World Airmail: 6-months £37.99, 12-months £70.99, 24-months £135.99

Cheques (in **£ sterling only**) payable to *Practical Electronics* and sent to: Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom

Tel: 01202 087631
Email: pesubs@selectps.com

Also via our online shop at**: www.electronpublishing.com**

Subscriptions start with the next available issue. We accept MasterCard or Visa.
(For past issues see the Back Issues page.)

DIGITAL SUBSCRIPTIONS

Electronic subscriptions are available from £24.99 for 12 months, more details at: **www.electronpublishing.com**

SUBSCRIPTION ORDER FORM

☐ 6 Months: UK £26.99, Europe £30.99 (Airmail), Rest of the World £37.99 (Airmail)
☐ 1 Year: UK £49.85, Europe £57.99 (Airmail), Rest of the World £70.99 (Airmail)
☐ 2 Years: UK £94.99, Europe £109.99 (Airmail), Rest of the World £135.99 (Airmail)

To: **Practical Electronics Subscriptions**
PO Box 6337
Bournemouth BH1 9EH
United Kingdom

Tel: 01202 087631
Email: pesubs@selectps.com

I enclose payment of £ ....
(cheque/PO in £ sterling only)
payable to *Practical Electronics*

☐ Please charge my Visa/Mastercard

My card number is: ....
Please print clearly, and check that you have the number correct

Card Security Code .... Valid From Date........................
(The last 3 digits on or just under the signature strip)

Card Ex. Date ....

Name ....

Address ....

Post code .... Tel. ....

Email ....

Subscriptions start with the next available issue.

STAY AT HOME!

AND READ

SILICON CHIP!

Of course we want you to continue to support Practical Electronics . . . but did you know you can also subscribe to SILICON CHIP – Australia's electronics magazine – direct from the United Kingdom (or anywhere else!)

And now there's a real choice:

(1) **A 12 month printed edition subscription** (12 issues), delivered right to your mailbox, for $AU150 per year (that's about £77 / €87 at time of publication)

(2) **A 12 month digital/online edition** (exactly the same content as the printed magazine) for $AU85 (about £44/ €50 at press time) – available to view when printed edition is released in Australia.

(3) **A 12 month combined printed and online subscription** – the best of both worlds – read now and have your printed edition available for future reference! That's just $AU170 for 12 months (about £87/ €98).

(4) **You can order a SILICON CHIP back issue** for $AU10 (about £5 / €6) plus P&P – most recent months still in stock

(5) **Now you can order SILICON CHIP archives on flash drive**: a quality metal flash drive containing any five year block of SILICON CHIP – from the first issue back in 1987 until Dec 2019 (see siliconchip.com.au/shop/digital_pdfs for details)

Log onto www.siliconchip.com.au for much more information!

You might also be interested in:

Radio, TV & Hobbies on DVD

Take a trip back in time for the entire Radio, TV and Hobbies magazine, from April 1939 through to March 1965 – ready to enjoy at your leisure, again and again and again.
Comes in a protective case – and it's just $AU62 plus p&p (about £31.60/€36)
Or you can download the digital edition of Radio, TV & Hobbies: $AU50 (£25.50 / €29)
See siliconchip.com.au/shop/3 for other options. And yes, you can order online!

SILICON CHIP

Australia's Electronics Magazine

DOWNLOAD YOUR PE/EPE BACK ISSUES

COMPLETE-YEAR collections – all 12 issues for each year!

From just £6.95 each

VOL 28 – January to December 1999 – **£6.95**
VOL 29 – January to December 2000 – **£6.95**
VOL 30 – January to December 2001 – **£6.95**
VOL 31 – January to December 2002 – **£6.95**
VOL 32 – January to December 2003 – **£6.95**
VOL 33 – January to December 2004 – **£6.95**
VOL 34 – January to December 2005 – **£6.95**
VOL 35 – January to December 2006 – **£6.95**
VOL 36 – January to December 2007 – **£6.95**
VOL 37 – January to December 2008 – **£6.95**
VOL 38 – January to December 2009 – **£8.95**
VOL 39 – January to December 2010 – **£8.95**
VOL 40 – January to December 2011 – **£8.95**
VOL 41 – January to December 2012 – **£12.95**
VOL 42 – January to December 2013 – **£12.95**
VOL 43 – January to December 2014 – **£14.95**
VOL 44 – January to December 2015 – **£14.95**
VOL 45 – January to December 2016 – **£16.95**
VOL 46 – January to December 2017 – **£18.95**
VOL 47 – January to December 2018 – **£22.95**
VOL 48 – January to December 2019 – **£24.95**
VOL 49 – January to December 2020 – **£25.95 – NEW!**

FIVE-YEAR collections – 60 issues!

From just £14.95 each

PE2832: Jan 99 to Dec 03 – £14.95
PE2933: Jan 00 to Dec 04 – £14.95
PE3034: Jan 01 to Dec 05 – £14.95
PE3135: Jan 02 to Dec 06 – £14.95
PE3236: Jan 03 to Dec 07 – £14.95
PE3337: Jan 04 to Dec 08 – £14.95
PE3438: Jan 05 to Dec 09 – £19.95
PE3539: Jan 06 to Dec 10 – £19.95
PE3640: Jan 07 to Dec 11 – £19.95
PE3741: Jan 08 to Dec 12 – £19.95
PE3842: Jan 09 to Dec 13 – £24.95
PE3943: Jan 10 to Dec 14 – £24.95
PE4044: Jan 11 to Dec 15 – £29.95
PE4145: Jan 12 to Dec 16 – £29.95
PE4246: Jan 13 to Dec 17 – £32.95
PE4347: Jan 14 to Dec 18 – £34.95
PE4448: Jan 15 to Dec 19 – £35.95 – NEW!

20 YEARS OF PE!

AMAZING BUNDLE OFFER
Every issue from 1999 – 2018

240 magazines for just

£49.95

PE2832: Jan 99 to Dec 03
PLUS!
PE3337: Jan 04 to Dec 08
PLUS!
PE3842: Jan 09 to Dec 13
PLUS!
PE3447: Jan 14 to Dec 18

Do you have the 15-year bundle?

Upgrade to the new 20-year bundle for just £9.95 with: PE15to20

Download your collections at: www.electronpublishing.com

Practical Electronics

Volume 50. No. 1
January 2021
ISSN 2632 573X

Editorial offices

Practical Electronics
Electron Publishing Limited
1 Buckingham Road
Brighton
East Sussex BN1 3RA

Tel 01273 777619
Mob 07973 518682
Fax 01202 843233
Email pe@electronpublishing.com
Web www.electronpublishing.com

Advertisement offices

Practical Electronics Adverts
1 Buckingham Road
Brighton
East Sussex BN1 3RA

Tel 01273 777619
Mob 07973 518682
Email pe@electronpublishing.com

Editor Matt Pulzer
General Manager Louisa Pulzer
Digital subscriptions Stewart Kearn **Tel** 01202 880299
Online Editor Alan Winstanley
Web Systems Kris Thain
Publisher Matt Pulzer

Print subscriptions

Practical Electronics Subscriptions
PO Box 6337
Bournemouth BH1 9EH
United Kingdom

Tel 01202 087631
Email pesubs@selectps.com

Technical enquiries

We regret technical enquiries cannot be answered over the telephone. We are unable to offer any advice on the use, purchase, repair or modification of commercial equipment or the incorporation or modification of designs published in the magazine. We cannot provide data or answer queries on articles or projects that are more than five years old.

Questions about articles or projects should be sent to the editor by email: pe@electronpublishing.com

Projects and circuits

All reasonable precautions are taken to ensure that the advice and data given to readers is reliable. We cannot, however, guarantee it and we cannot accept legal responsibility for it.

A number of projects and circuits published in *Practical Electronics* employ voltages that can be lethal. You should not build, test, modify or renovate any item of mains-powered equipment unless you fully understand the safety aspects involved and you use an RCD (GFCI) adaptor.

Component supplies

We do not supply electronic components or kits for building the projects featured, these can be supplied by advertisers. We advise readers to check that all parts are still available before commencing any project in a back-dated issue.

Advertisements

Although the proprietors and staff of *Practical Electronics* take reasonable precautions to protect the interests of readers by ensuring as far as practicable that advertisements are bona fide, the magazine and its publishers cannot give any undertakings in respect of statements or claims made by advertisers, whether these advertisements are printed as part of the magazine, or in inserts. The Publishers regret that under no circumstances will the magazine accept liability for non-receipt of goods ordered, or for late delivery, or for faults in manufacture.

Transmitters/bugs/telephone equipment

We advise readers that certain items of radio transmitting and telephone equipment which may be advertised in our pages cannot be legally used in the UK. Readers should check the law before buying any transmitting or telephone equipment, as a fine, confiscation of equipment and/or imprisonment can result from illegal use or ownership. The laws vary from country to country; readers should check local laws.

Editorial

Looking forward to 2021

Where to begin with the year 2020? It has certainly been a year we'll never forget: lockdowns, endless statistics, economic uncertainty and sickness. While it's no antidote to a pandemic, I do hope *Practical Electronics* has provided a welcome distraction from some very serious problems. With luck, and a huge amount of brilliant work from dedicated researchers, 2021 may well bring a vaccine and the chance to put the virus behind us.

In the meantime, I hope you enjoy this month's *Practical Electronics*, which marks the very welcome return of *PE*/*EPE* stalwart Mike Tooley. He is starting a new series – *KickStart* – which aims to cover a wide range of topics with useful down-to-earth advice, tips and hints.

For those of you who enjoy building Hi-Fi with valves we have a special treat in the shape of the *Nutube Miniature Valve Stereo Preamplifier*. This is based around a fascinating new low-voltage thermionic valve technology that we hope to use again in *PE*.

Thank you, PE writers!

Practical Electronics stands or falls with the quality of its writing. I always end a year with a thank you to our contributors. They work very hard every month to bring you high-quality, original content. So, in no particular order – a great big 'well done and thank you' to Alan Winstanley, Mike Tooley, Ian Bell, Mark Nelson, Mike Hibbett, Clive 'Max' Maxfield, Phil Boyce, Julian Edgar, Barry Fox and Jake Rothman. Just as important are the 'back office boys', Stewart Kearn, (again) Alan Winstanley and Kris Thain, who keep the shop and website ticking over.

Extra special Christmas present?

Even – or perhaps especially – a socially distanced holiday season still comes with the unavoidable 'What would you like for Christmas?'. For you though, lucky reader, the answer is easy – 'I just want a subscription to my favourite magazine.' You can choose paper or online, and as a subscriber you can be sure that you will never miss a copy.

From all of us at *Practical Electronics*, thank you for your support over 2020, have a very happy Christmas and a healthy 2021!

Matt Pulzer

Publisher

The **Fox** Report

Barry Fox's technology column

HDMI capture dongle update

I promised an update report on the absurdly low-priced (under a tenner) HDMI capture dongle which I test-purchased (see *The Fox Report*, *PE*, October 2020.) These little devices are now widely available online, direct from Chinese suppliers and via more local importers – at a wide range of prices starting at around £7 (US$10).

The generic name is 'HDMI to USB Video Capture Card 1080P 2.0 For Game / Live Streaming'. Some middlemen are labelling these dongles with their own model names, but the basic specification, and very likely the chipset, remains the same:

'Input (HDMI) resolution 3840×2160@30Hz, output (USB) resolution 1920×1080@30Hz. Support 8/10/12-bit deep colour. Support most acquisition software, such as VLC, OBS, Amcap, etc. Support Windows, Android and macOS. No external power supply is required. Video Output Mode: YUV / JPEG. Support Audio Format: L-PCM'.

Super-cheap HDMI capture dongle – it does exactly what it says on the case...

Output
HDMI®
HIGH-DEFINITION MULTIMEDIA INTERFACE
Video Capture
Input

...but unlike its (much) more expensive rivals it does require a reasonably powerful PC.

Easy to use

No instructions are needed. Just plug one end of the dongle (which is around the same size as a USB memory device) into the HDMI output of pretty much any media player, set-top box, game console or online entertainment system; connect the other end of the dongle by USB lead to a USB socket on a computer which is running capture software, such as free Open Source OBS (Open Broadcaster Software).

The capture software 'sees' the dongle as a simple USB media device and captures whatever it delivers. It's that simple. The only complication is the way OBS offers a myriad of setup options and likes to default to settings which mute the sound. Time spent on the OBS help pages and googling user experience is time well spent. It's a small price to pay for a wonderfully flexible free program.

Higher-priced alternatives

One cautionary note; most of the capture devices sold by big name companies such as Hauppauge and Elgato, and costing ten or twenty times the dongle price, have on-board processing power which takes processing load off the computer running the capture software (such as OBS). This enables glitch-free capture on low-spec computers. Some of these 'name' devices also cater for component video and analogue and SP/DIF audio inputs; the new dongles do not.

If you use the budget dongle with a budget computer, the captured video and audio will stutter and judder. But use it with a reasonably powerful computer and the dongle delivers smooth and glitch-free results. I tried it with a low-end Acer Windows 10 laptop and ageing HP desktop and the results were unacceptable; but with a Mac laptop it worked like a dream.

HAMMOND
MANUFACTURING®

new!

1552 hand-held plastic enclosures

Learn more: hammfg.com/1552

Contact us to request a free evaluation sample.
uksales@hammfg.com • 01256 812812

Copyright / copy wrong

Tests suggest that the dongle simply ignores the HDCP copy protection which is tied to the HDMI standards; so, it enables copyright infringement. Perhaps sales can and will be banned; perhaps not. We are simply reporting the facts on devices which are now widely on sale, with China apparently the prime source.

What you use these dongles for is your responsibility. I have found an HDMI-to-USB dongle a very good way of digitising my own old home videos, by first converting analogue AV from a VCR to HDMI (see *The Fox Report*, *PE*, September 2020) with one of the equally cheap converter devices that are now also widely available. The analogue-to-HDMI conversion will often stabilise ageing video.

OUTPUT
SCART ↔ HDMI
Scaler Converter
INPUT
SCART
SCART ↔ HDMI
Scaler Converter
OUTPUT
1080P/720P
HDMI
Power
DC/5V

VCR rescue! Combine an HDMI capture dongle with one of the many cheap and simple SCART-to-HDMI converters to get your tape-based analogue video into the digital domain.

AVAILABLE NOW!

PE VOL 49
January to December 2020

PDF files ready for immediate download

All 12 2020 issues for just £25.95

See page 6 for further details and other great back-issue offers.

Purchase and download at:
www.electronpublishing.com

STEWART OF READING
17A King Street, Mortimer, near Reading, RG7 3RS
Telephone: 0118 933 1111 Fax: 0118 933 2375
USED ELECTRONIC TEST EQUIPMENT
Check website www.stewart-of-reading.co.uk
(ALL PRICES PLUS CARRIAGE & VAT)
Please check availability before ordering or calling in

Fluke/Philips PM3092 Oscilloscope
2+2 Channel 200MHz Delay TB,
Autoset etc – £250

HP 54600B Oscilloscope
Analogue/Digital Dual Trace 100MHz
Only £75, with accessories £125

LAMBDA GENESYS	PSU GEN100-15 100V 15A Boxed As New	£400
LAMBDA GENESYS	PSU GEN50-30 50V 30A	£400
IFR 2025	Signal Generator 9kHz – 2.51GHz Opt 04/11	£900
IFR 2948B	Communication Service Monitor Opts 03/25 Avionics	POA
IFR 6843	Microwave Systems Analyser 10MHz – 20GHz	POA
R&S APN62	Syn Function Generator 1Hz – 260kHz	£295
Agilent 8712ET	RF Network Analyser 300kHz – 1300MHz	POA
HP8903A/B	Audio Analyser	£750 – £950
HP8757D	Scaler Network Analyser	POA
HP3325A	Synthesised Function Generator	£195
HP3561A	Dynamic Signal Analyser	£650
HP6032A	PSU 0-60V 0-50A 1000W	£750
HP6622A	PSU 0-20V 4A Twice or 0-50V 2A Twice	£350
HP6624A	PSU 4 Outputs	£400
HP6632B	PSU 0-20V 0-5A	£195
HP6644A	PSU 0-60V 3.5A	£400
HP6654A	PSU 0-60V 0-9A	£500
HP8341A	Synthesised Sweep Generator 10MHz – 20GHz	£2,000
HP83630A	Synthesised Sweeper 10MHz – 26.5 GHz	POA
HP83624A	Synthesised Sweeper 2 – 20GHz	POA
HP8484A	Power Sensor 0.01-18GHz 3nW-10μW	£75
HP8560E	Spectrum Analyser Synthesised 30Hz – 2.9GHz	£1,750
HP8563A	Spectrum Analyser Synthesised 9kHz – 22GHz	£2,250
HP8566B	Spectrum Analsyer 100Hz – 22GHz	£1,200
HP8662A	RF Generator 10kHz – 1280MHz	£750
Marconi 2022E	Synthesised AM/FM Signal Generator 10kHz – 1.01GHz	£325
Marconi 2024	Synthesised Signal Generator 9kHz – 2.4GHz	£800
Marconi 2030	Synthesised Signal Generator 10kHz – 1.35GHz	£750
Marconi 2023A	Signal Generator 9kHz – 1.2GHz	£700

Marconi 2305	Modulation Meter	£250
Marconi 2440	Counter 20GHz	£295
Marconi 2945/A/B	Communications Test Set Various Options	POA
Marconi 2955	Radio Communications Test Set	£595
Marconi 2955A	Radio Communications Test Set	£725
Marconi 2955B	Radio Communications Test Set	£800
Marconi 6200	Microwave Test Set	£1,500
Marconi 6200A	Microwave Test Set 10MHz – 20GHz	£1,950
Marconi 6200B	Microwave Test Set	£2,300
Marconi 6960B	Power Meter with 6910 sensor	£295
Tektronix TDS3052B	Oscilloscope 500MHz 2.5GS/s	£1,250
Tektronix TDS3032	Oscilloscope 300MHz 2.5GS/s	£995
Tektronix TDS3012	Oscilloscope 2 Channel 100MHz 1.25GS/s	£450
Tektronix 2430A	Oscilloscope Dual Trace 150MHz 100MS/s	£350
Tektronix 2465B	Oscilloscope 4 Channel 400MHz	£600
Farnell AP60/50	PSU 0-60V 0-50A 1kW Switch Mode	£300
Farnell XA35/2T	PSU 0-35V 0-2A Twice Digital	£75
Farnell AP100-90	Power Supply 100V 90A	£900
Farnell LF1	Sine/Sq Oscillator 10Hz – 1MHz	£45
Racal 1991	Counter/Timer 160MHz 9 Digit	£150
Racal 2101	Counter 20GHz LED	£295
Racal 9300	True RMS Millivoltmeter 5Hz – 20MHz etc	£45
Racal 9300B	As 9300	£75
Solartron 7150/PLUS	6½ Digit DMM True RMS IEEE	£65/£75
Solatron 1253	Gain Phase Analyser 1mHz – 20kHz	£600
Solartron SI 1255	HF Frequency Response Analyser	POA
Tasakago TM035-2	PSU 0-35V 0-2A 2 Meters	£30
Thurlby PL320QMD	PSU 0-30V 0-2A Twice	£160 – £200
Thurlby TG210	Function Generator 0.002-2MHz TTL etc Kenwood Badged	£65

HP33120A	Function Generator 100 microHz – 15MHz	£350
HP53131A	Universal Counter 3GHz Boxed unused	£600
HP53131A	Universal Counter 225MHz	£350
Audio Precision	SYS2712 Audio Analyser – in original box	POA
Datron 4708	Autocal Multifunction Standard	POA
Druck DPI 515	Pressure Calibrator/Controller	£400
Datron 1081	Autocal Standards Multimeter	POA
ENI 325LA	RF Power Amplifier 250kHz – 150MHz 25W 50dB	POA
Keithley 228	Voltage/Current Source	POA
Time 9818	DC Current & Voltage Calibrator	POA

HP/Agilent HP 34401A Digital
Multimeter 6½ Digit £325 – £375

Marconi 2955B Radio
Communications Test Set – £800

Bad boy bulbs

Techno Talk

Mark Nelson

Unwelcome interference is again our topic this month, but on a different kick. This time, the investigation team performed splendidly, tracking down a hapless householder who was jeopardising the safety of the 8.85 million passengers who use Glasgow airport each year.

This month's EMR nightmare starts with the oldest, and you'd think humblest of electrical items – the lightbulb. Someone unwittingly bought some light bulbs that should never have been put on sale in the first place, and which had defects that were recognised 70 years ago.

Mission impossible?

Pilots flying aircraft in and out of Glasgow airport complained they were suffering radio blackouts when they were between 6,000 and 10,000 feet in the air.

Voice communication between aircrew and the controllers on the ground was wiped out by mystery interference, meaning crews were unable to hear vital air traffic control messages. Locating and identifying the source of the jamming signal aloft was well-nigh impossible on account of the height of the aircraft and their flying speed. Finding the source was a veritable needle-in-a-haystack puzzle, but OFCOM's investigators were not easily deterred.

An 'area of probability' was narrowed down by using flight-tracking software to correlate the location of planes reporting the issue with the corresponding location on the ground. With this done, the quest turned into a ground-level investigation centred on a single town. This task involved using receivers aboard vehicles and driving through the suspected area until the interference was heard.

Old-fashioned footwork

Having pinned down the area where the signal was strongest, the investigators then used handheld equipment to cover the remainder of the search area on foot. The team visited a number of likely properties and they eventually located the source. But what was jamming the sensitive aircraft radios so effectively?

If you're a regular reader of this column (and if not, why not?), you may well be screaming: 'Oh no, it's those blessed LED light bulbs that hoot from DC to daylight', but you'd be wrong. In fact, the culprits were squirrels – or rather their cages. Let me explain, albeit in a roundabout way.

The problem with squirrels

First, let's recall a very apposite statement made in 1905 by the philosopher George Santayana in his book, *The Life of Reason*. It read, 'Those who cannot remember the past are condemned to repeat it.' The radio jamming at Glasgow airport proves his point perfectly.

Back in 1905, the incandescent lamp bulb was not particularly efficient and to counter this, manufacturers were doing their damnedest to squeeze the maximum amount of light from a glowing filament. One technique involved threading the wire filament up and down the length of the bulb multiple times in a circular fashion using a system of hooks and frames until it resembled a squirrel cage. In those days, keeping pet squirrels in cages was not uncommon, although nowadays it's considered pretty cruel by most people.

After the sale of most incandescent bulbs was phased out by European legislation, lamp manufacturers discovered a loophole that allowed them to sell 'decorative' incandescent lamps, and the squirrel cage bulb has returned. These are usually made of tinted glass and have the word 'rustic' in their description; for example: **https://amzn.to/32C1BTL**

Get to the point!

OK, so what's the connection between radio jamming and Santayana? Precisely this – OFCOM's photo of the errant lamp bulb that was wiping out air traffic control frequencies (118 to 121MHz, remember this!) showed it was a squirrel cage 'deco' bulb. Electromagnetic radiation does not occur with modern incandescent bulbs that use short, coiled filaments, but the multiple straight wires used in squirrel cage lamps radiate like mad and can affect FM radio transmissions (88 to 108MHz) and the nearby air band (118 to 121MHz). The mechanism is so deeply technical that I cannot paraphrase it in a sentence, but you can read about it here: **https://bit.ly/pe-jan21-fm**

Back in the early 1950s this potential for harmful interference was already recognised in America, where power companies were changing out straight-filament bulbs for consumers (read *Popular Science* magazine, April 1953, **https://bit.ly/pe-jan21-ps**). Nobody cares today, though, because those who do not remember the past are condemned to repeat it.

Happy ending?

OFCOM confirms that the interference at Glasgow was caused by four 'vintage' lightbulbs that the homeowner had recently bought online (not necessarily the ones mentioned above). The house was directly underneath the flightpath of the aircraft and so every time a plane passed and the lamps were in use, the air crew suffered the interference. The bulbs were replaced with non-interfering types and checks with National Air Traffic Services and aircraft operators confirm that the area is now free of interference. OFCOM's spectrum enforcement team is following up the case with the lightbulb suppliers, to make sure no more dangerous bulbs are sold to unwitting customers.

Tracking down the disturbance that was not merely annoying but positively life-threatening was a highly impressive effort. This work, by telecoms regulator OFCOM, confers great credit on their spectrum assurance team, who found the unknowing culprit.

So that's all right then? Hardly. Squirrel cage lamp bulbs are still sold widely; as I write this article, eBay has 1,284 offers of them for sale, while Amazon has more than 2,000. All manner of other online sellers and out-of-town DIY sheds sell them as well. How many of those bulbs have been tested for EMC compatibility? Manufacturers – and UK distributors – are responsible for exercising due diligence regarding the CE marking they place on their products, but how many of them are aware of their legal responsibilities for public safety? And what if they are made on the other side of the world, where CE merely signifies 'Chinese export'? Local trading standards authorities are responsible for enforcing regulations, but they may not be aware of non-conformant merchandise. So, as always – *caveat emptor* – let the buyer beware!

Exclusive offer

Win a Microchip SAM IoT WG Development Board

Practical Electronics is offering its readers the chance to win a SAM IoT WG Development Board (EV75S95A) – and even if you don't win, receive a 20%-off voucher, plus free shipping for one of these products.

The SAM-IoT WG Development Board features the SAMD21G18 Arm Cortex-M0+ based 32-bit microcontroller (MCU), an ATECC608A CryptoAuthentication secure element IC and the fully certified ATWINC1510 Wi-Fi network controller, so you can quickly and easily connect your embedded application to Google's Cloud IoT core platform.

The on-board debugger allows you to program and debug the MCU without any additional hardware. Use the mikroBUS sockets to expand your design with your choice of MikroElekronika click Boards. The SAM-IoT WG development board is a small and easily expandable demonstration and development platform for IoT solutions.

Out of the box, the MCU comes preloaded with firmware that enables you to quickly connect and send data to the Google Cloud Platform. Once you are ready to build your own custom design, you can easily generate code using the free software libraries in MPLAB Harmony v3. The SAM-IoT WG Development Board is supported by MPLAB X IDE.

The SAM-IoT WG development board features two sensors:

- A light sensor – TEMT6000
- A high-accuracy temperature sensor – MCP9808

Worth $29.00 (approx £22.50) each

Free-to-enter competition

Microchip
SAM IoT WG Development Board

How to enter

For your chance to win a Microchip SAM IoT WG Development Board or receive a 20%-off voucher, including free shipping, enter your details in the online entry form at:

https://page.microchip.com/PE-SAM-IoT.html

Closing date

The closing date for this offer is 31 December 2020.

September 2020 winner

Gary Meakin

He won a Microchip Explorer 16/32 Development Kit

Net Work

Alan Winstanley

This month, Net Work has some of the latest news on electric cars, tips and tricks for running an Android-based TV system and advice on setting up a two-screen Windows PC.

In last month's Net Work, I discussed some of the trends appearing in the electric vehicle (EV) market, starting with the now Chinese-owned 'MG' marque, a re-emerging former British car brand that has ambitions for growth in the UK and Europe. After introducing a small range of petrol cars, MG Motors is busily positioning itself as makers of all-electric battery (BEV) and plug-in hybrid (PHEV) cars, as demonstrated in a recent TV advertising campaign for the MG ZS EV that I mentioned last month (**see https://youtu.be/mAqeTrA8I3c**). We Brits also like our estate cars (station wagons), and MG has launched the MG5 SW, a new 52.5kWh EV station wagon with a range of (up to) 214 miles. The brand is also launching in Ireland and aims to expand further across Europe and Australia, backed by the resources of China's SAIC motor group. For everyday consumers, MG's emphasis will be on affordability, promising an era of 'electric for everyone'.

The zany Honda 'e' (see last month) is an astonishing mini town car bristling with LCDs and electronics aimed purely at urban drivers. Readers can find an engaging review of Honda's baby battery car (and its interactive LCD aquarium!) on YouTube at **https://youtu.be/x6G-3_Aasao**

The same channel also reviewed Volkswagen's forthcoming ID.3 all-electric car, VW's third iteration of its 'world car' following the VW Beetle and Golf. This '1st Edition' BEV costs nearly £39,000 ($50,000) before government grants (cheaper ones are in the pipeline) and VW reportedly expects to sell three million electric vehicles a year within four years, so expect them (and similar SEAT brand cars) to become a common sight before the decade is out.

See **https://youtu.be/4yntwHNCDBo** for an early review, and details of VW electric cars can be found on VW's web site at: **www.volkswagen.co.uk/electric**

Electric car WLTP...

How to measure fuel economy of an electric car? For those who, like the author, still check their miles per gallon, a new set of rules has been adopted that covers both new petrol/diesel cars and EVs. The term 'WLTP' (World harmonised Light-vehicle Test Procedure) is cropping up everywhere, including electric vehicle marketing which aims to give buyers real-world fuel consumption of vehicles. We might have to re-think our motoring habits and get used to the idea of journeys being interrupted by a 30 or 40-minute break to 'refuel' our plug-in cars at a (hopefully) nearby charging point. More advice about WLTP data can be found on the UK government website at: **https://bit.ly/pe-jan21-wltp**

It is still early days for electric vehicles as new petrol, diesel and hybrid sales are not scheduled to be banned in the UK until the year 2030. Until then, many drivers might prefer to hang on and let the BEV market mature. An emerging trend is that of owners facing potentially ruinous repair bills once their car warranties have expired. Worrying reports circulate of £40,000 hybrid cars suffering major electric failures after just three years, leaving owners with nearly £3,000 in repair bills from workshops unfamiliar with this new technology and car manufacturers reluctant to pitch in to help.

Android on TV

Set-top box maker Humax has launched the first 4K Freeview Play recorder that also runs Android TV. The feature-filled Humax Aura has a 1TB or 2TB hard drive and can pause or play live TV as well as recording 4K HDR programmes, plus the Aura also runs Android TV 9, which gives access to some 5,000 apps on Google Play Store. Freeview Play offers built-in catch-up TV for mainstream TV channels and it can also record up to four programmes at a time while watching a fifth. Parents might welcome the Kids Zone feature and the remote carries Google Assistant and Prime Video buttons. Its built-in Google Assistant interacts in the usual way and allows smart control of compatible IoT devices, while the Chromecast-like operation allows content on other compatible devices to be cast to the Humax Aura.

A not-yet-available Aura Android app promises to allow live or recorded content to be streamed around the house and reminders to be set. Assuming the app will be fully supported in

VW's forthcoming ID.3 BEV aims to be easy to live with and brings affordable electric motoring to the masses.

The VW's dashboard has a central display and smaller screens in front of the driver. Note the 'Play' and 'Pause' pedals.

The Humax Aura is a 4K personal video recorder complete with Freeview Play and Android TV. 1TB and 2TB disk capacities are available.

the future, the Aura could be a worthwhile way of adding Android PVR functionality to a 4K or HD TV. The recorder costs £249 or £279 depending on disk capacity. More details at: **https://bit.ly/pe-jan21-aura**

If a fully-fledged 4K Android TV recorder is not for you, Google has finally launched its 'Chromecast with Google TV', a 4K dongle with voice remote control sporting Google Assistant, Netflix and YouTube buttons (see *Net Work*, November 2020). It adds Chromecast-style Android TV services to any TV with an HDMI port, which may give an older TV a new lease of life. All the usual streaming services are available, including Amazon Prime, mainstream catch-up TV, Google apps and, of course, Netflix and YouTube. The dongle aims to make it easy to cast content from other Android and iOS devices. Where Wi-Fi is weak, an Ethernet cable can be connected to a special mains-powered USB adaptor, available separately. The 'Chromecast with Google TV' debuted on YouTube at: **https://youtu.be/EPluWn8RT3Y** and can be purchased for £59.99 at **store.google.com**, but expect special offers to be available.

Amazon users might choose from their latest line-up of Fire TV Stick HDMI dongles and several models are now available in time for Christmas: the cheapest 'Lite' version (£29.99) has an Alexa voice remote and the mid-range TV Stick (£39.99) adds TV volume and power buttons which (only) work with 'compatible' TVs and soundbars, Amazon says. A more powerful 4K UHD version (£49.99) is also sold, and if your Wi-Fi coverage is patchy then wired Ethernet adaptors are sold separately.

Chromecast with Google TV offers 4K connectivity with Android TV features and a voice remote control.

Amazon's Fire TV Cube has a built-in speaker and Alexa commands interface with your TV screen.

For hands-free control with a single TV, Amazon's Fire TV Cube might be the answer. It has a built-in speaker offering the usual Alexa interface, displaying on the TV screen as well. The Fire TV Cube costs £109.99 and claims to control certain smart devices and selected 'compatible' Sky satellite receivers too, but do check the specs carefully. Full details are on **Amazon.co.uk** and do keep a close eye on prices and reviews: these can drop wildly over a holiday weekend so set up the Camelizer plug-in to alert you of price drops, see: **https://camelcamelcamel.com/camelizer**

Twin-dows 10!

After upgrading a work PC to Windows 10, a 22-inch monitor was left over that still had plenty of life in it. What to do with a surplus screen that was too good to throw away? Provided your PC motherboard or video card has a suitable spare video port, Windows 10 can make short work of adding a second monitor to allow dual-screen operation and increase your desktop's 'real estate'. After hooking up a second screen, simply hit [Windows key] + P to 'project' your desktop on to the second monitor, choosing the 'Extend' mode from the menu. If your main monitor goes blank, keep holding the Windows key and hit 'P' to cycle through the options.

Dual-screen operation has proved very handy for a busy desktop worker, with the extra screen space great for dragging active windows or website shortcuts sideways out of the way. Using a PC TV tuner, such as a Hauppauge USB type, means you can drop a TV window onto the second monitor or you could watch Freeview TV in a web browser instead; go to: **https://bit.ly/pe-jan21-free** and choose 'watch now' or 'catch up', then drop the new TV browser window onto the second monitor. This idea works very well, though the odd freeze or lockup has occurred in practice.

The author's ordinary video card drives both a DVI and VGA monitor in dual-screen mode without any problem. Spare (or lengthier) video cables are readily available online, as are wireless Wi-Fi display dongles (untested by the author) that enable PC or mobile device screens to be mirrored on the monitor from a distance. Thanks to the simple setup in Windows 10 it's easy to build a dual-screen option, so if you have a spare monitor gathering dust why not test it on a spare video port and see if dual-screen mode will work for you.

Some other handy keyboard shortcuts worth knowing include:

`Win + P`	dual monitors
`Win + V`	clipboard history
`Win + I`	PC system settings
`Win + Tab`	30-day timeline
`Win + D`	toggle Windows desktop
`Win + E`	File Explorer
`Win + L`	Lock your PC
`Win + R`	Open the Run... dialogue
`Win + Pause/Break`	About page
`Ctrl + Shift + Esc`	Task Manager

A list of Windows keyboard shortcuts is on Microsoft's website: **https://tinyurl.com/y5o7pwp2** (some are now obsolete).

Archiving disks

One final tip: after upgrading the aforementioned PC, like many users I was left with a handful of hard disks that contained valuable legacy data, files and photos, so I archived the drives safely in plastic HDD storage boxes sourced on eBay (eg, **https://tinyurl.com/y53jtngp**). For many years I've relied on a handy cataloguing program called SuperCat which can index entire hard disks or data DVDs and catalogue them in a searchable database. I can easily search the catalogue on-screen in order to pinpoint the relevant hard drive. A 30-day trial of this neat little program (32/64-bit, Windows 10 compatible, $35) can be downloaded from **http://no-nonsense-software.com/supercat**

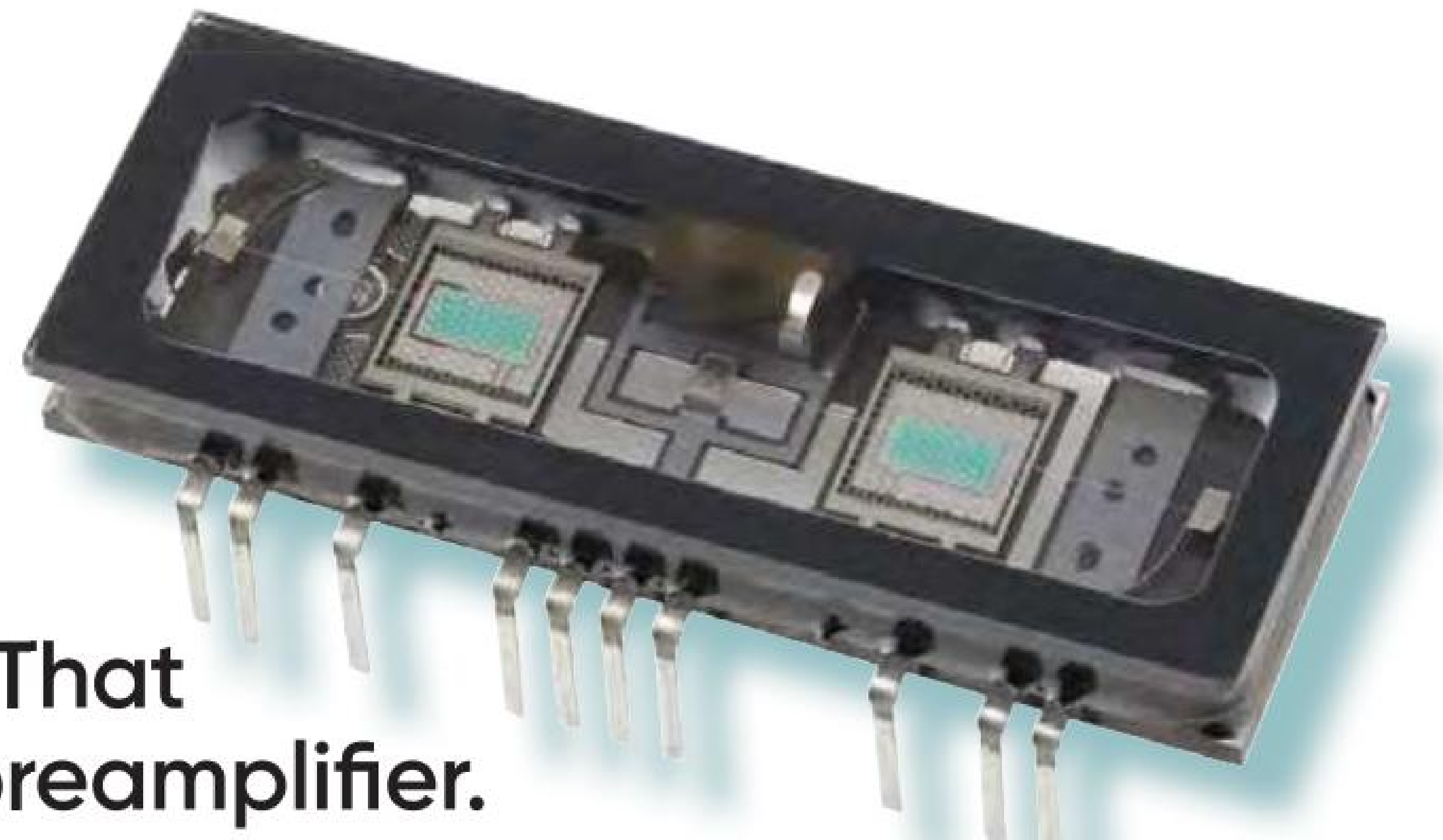
Wishing all our readers a Happy Christmas and a safer 2021.

The author can be reached at:
alan@epemag.net

Nutube miniature valve stereo preamplifier

by John Clarke

Valves are old hat, right? Not any more, they're not! Korg and Noritake Itron of Japan recently released their Nutube 6P1 twin triode. Its party trick is a very wide range of operating voltages, from just a few volts up to 200V, and meagre power consumption. That makes it ideal for a battery-powered stereo preamplifier. You'll enjoy the sound as well as the retro blue glow!



Are you one of those people who simply 'loves' the nostalgic sound of valves, both in power amplifiers and preamps? But valves are relatively expensive, and the high-voltage power supplies typically required make building a valve preamp a bit of a pain.

Now, however, the latter problem is no longer true with Korg's Nutube 6P1 twin-triode. It works perfectly fine with a plate voltage of just 6-12V, and the heater power and voltage requirements are also modest.

So, building a preamp around it is a cinch, and it's a suitable project for beginners and school students becuase there are no dangerous voltages involved. (In fact, for this reason alone we anticipate that this will be a very popular student project, right up to and including their 'major work').

Even if you have built valve gear with high voltage supplies before, we think you will find the unusual construction of the Nutube 6P1 dual triode quite fascinating.

We've taken some care with this design, so that it fits into a very cool (and professional) looking extruded aluminium case, with the inputs and outputs at the rear and a power switch and volume knob at the front. And of course, we've left a window in the clear front panel so that you can see that 'warm' blue tube glow.

One of the fascinating aspects of the Nutube is that it's designed and built similarly to a vacuum fluorescent display (VFD). So the heater glow looks like two blue squares, similar to large VFD pixels.

Its performance is pretty good, too. Distortion levels below 0.1% are possible across a wide range of frequencies with a little care during calibration. See the spec panel, Fig.2 and Fig.3, plus Fig.12 to get an idea of how well it performs.

This Nutube preamp can run from a DC supply between 7V and 18V, with only a modest current draw. It can also be powered using a 9V battery that is housed within the preamp's enclosure.

If you want to be able to switch between signal sources, you can match up this *Nutube Preamplifier* with the *Six-way Stereo Audio Input Selector with Remote Control* that we described in the June 2020 issue.

Nutube 6P1 dual triode

Korg developed the Nutube 6P1 in collaboration with Noritake Itron of Japan. While it is a directly heated triode with a filament, grid and plate connections, its construction more resembles a vacuum fluorescent display (VFD) than a traditional valve (or tube).

Two Nutube triodes are encapsulated in a rectangular glass envelope. Each triode is effectively a single-

Specifications

- Power supply: 7-18 VDC; draws 29mA @ 9V DC
- Gain: up to 15dB at maximum volume setting
- Distortion: around 0.07% at 200mV RMS output from 20Hz to 5kHz (see Fig.1 and Fig.2)
- Frequency response: 20Hz-20kHz, +0,–0.6dB; –3dB at about 7Hz and 80kHz (see Fig.3)
- Channel separation: typically >45dB (see Fig.4)
- Signal-to-noise ratio: 83dB with respect to 270mV in, 2V out, 20Hz-22kHz bandwidth
- Maximum output level: 2V RMS with 9V supply, 2.8V RMS with 12V supply

Features

- Stereo valve preamplifier
- Based on the recently released Korg 'Nutube' dual triode
- Visible plate glow
- 30,000-hour Nutube life
- *Safe* low-voltage supply (7-18V DC)
- Low power consumption
- Battery or plugpack powered
- Onboard volume control
- Internal balance and distortion adjustments
- Switch-on and switch-off noise eliminated
- Power supply reverse-polarity protection
- No transformers needed
- Inputs and outputs are in phase.

pixel VFD. The internal construction has the heater filament as a fine-gauge wire running across the front, with the metal mesh grid located below that. Behind the grid is the plate (also called the anode), which is phosphor-coated and glows when the filament is heated.

The filament wire is held taut, and because of this, it can vibrate similarly to a stringed musical instrument. (The Nutube is, after all, sold by a musical instrument manufacturer).

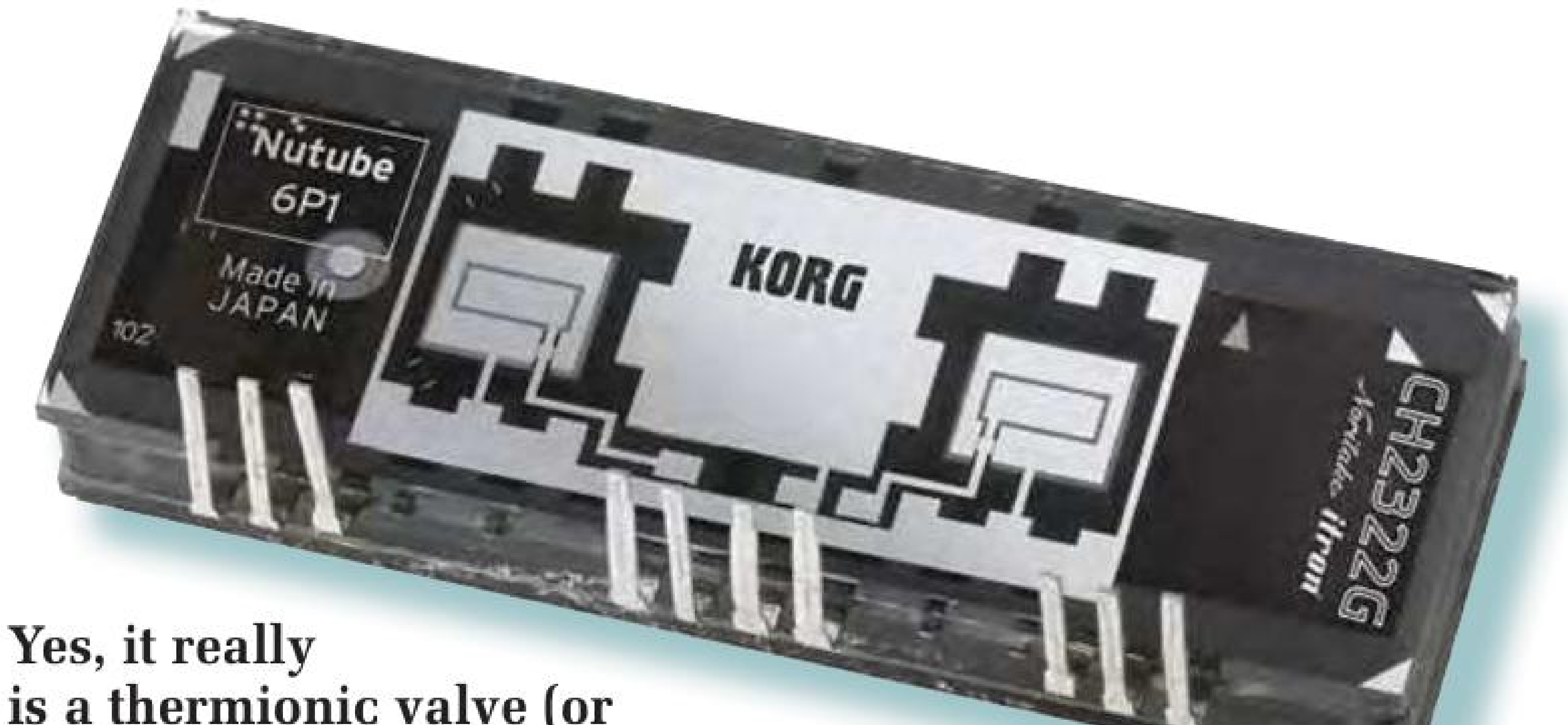
This vibration is not necessarily a wanted feature, as it can be the source of microphonics – where external sound can couple to the filament and this alters (or modulates) the audio signal being amplified in the triode. The result is that this vibration is heard in the sound output.

The microphonics can be minimised using careful construction methods. This includes protecting the Nutube from surrounding air vibrations, by using flexible wiring and including a vibration-damped mounting.

In operation, the Nutube draws very little current, with each filament requiring just 17mA. Total heater power for the two triodes is around 25mW. The grid and plate current total at around 38µA.

The Nutube is best operated with a plate voltage between 5V and 30V, and the load-line curves (Fig.1) reveal that within this voltage range, the grid voltage needs to be above the cathode filament.

This is different from the traditional triode, where plate voltages are much higher, and the grid voltage is usually negative with respect to the cathode.



Yes, it really is a thermionic valve (or 'tube' as our Americans friends like to say). But this Nutube 6P1, shown here from the underside, is quite unlike any valve you've come across before. For a start, those blue windows (see opposite) really do glow blue!

The Nutube operating point would typically be set so that the distortion from each triode is at a minimum and so that maximum dissipation is not exceeded. To achieve this, our design includes two trimpots to set the grid bias of each triode.

There are three ways to make these adjustments. One is to adjust the trimpots so the Nutube plate glows brightest for each channel, which will generally give good performance.

Another method is to use a signal source and multimeter to adjust the grid bias for maximum output signal level, or better still, by observing the distortion products and setting each trimpot for the desired result.

Freely available computer software can be used to measure the distortion and view the waveform. This allows for easy set-up of the desired distortion characteristic. We describe what software you need and how to use it in a panel later in this article.

Preamplifier performance

Fig.2 shows the total harmonic distortion plus noise (THD+N) figure as a percentage, plotted against frequency and output level. As you can see from Fig.3, the performance is best with an output level in the 100-400mV RMS range.

This is a typical level that you might feed into a 100W (or thereabouts) stereo amplifier to get a reasonable listening volume. Such an amplifier would generally have a full -power sensitivity between 1-2V RMS.

Below 100mV RMS output, noise starts to dominate the THD+N figure. In other words, preamp performance at lower volume levels is limited by its 83dB ultimate signal-to-noise ratio (SNR). Above 400mV RMS, triode non-linearities dominate.

SILICON CHIP Nutube Preamplifier
Volume
Power

Tiny! At just 115 × 50 × 125mm, and built into this snazzy extruded case from Jaycar, it really looks the part. Performance is no slouch, either!

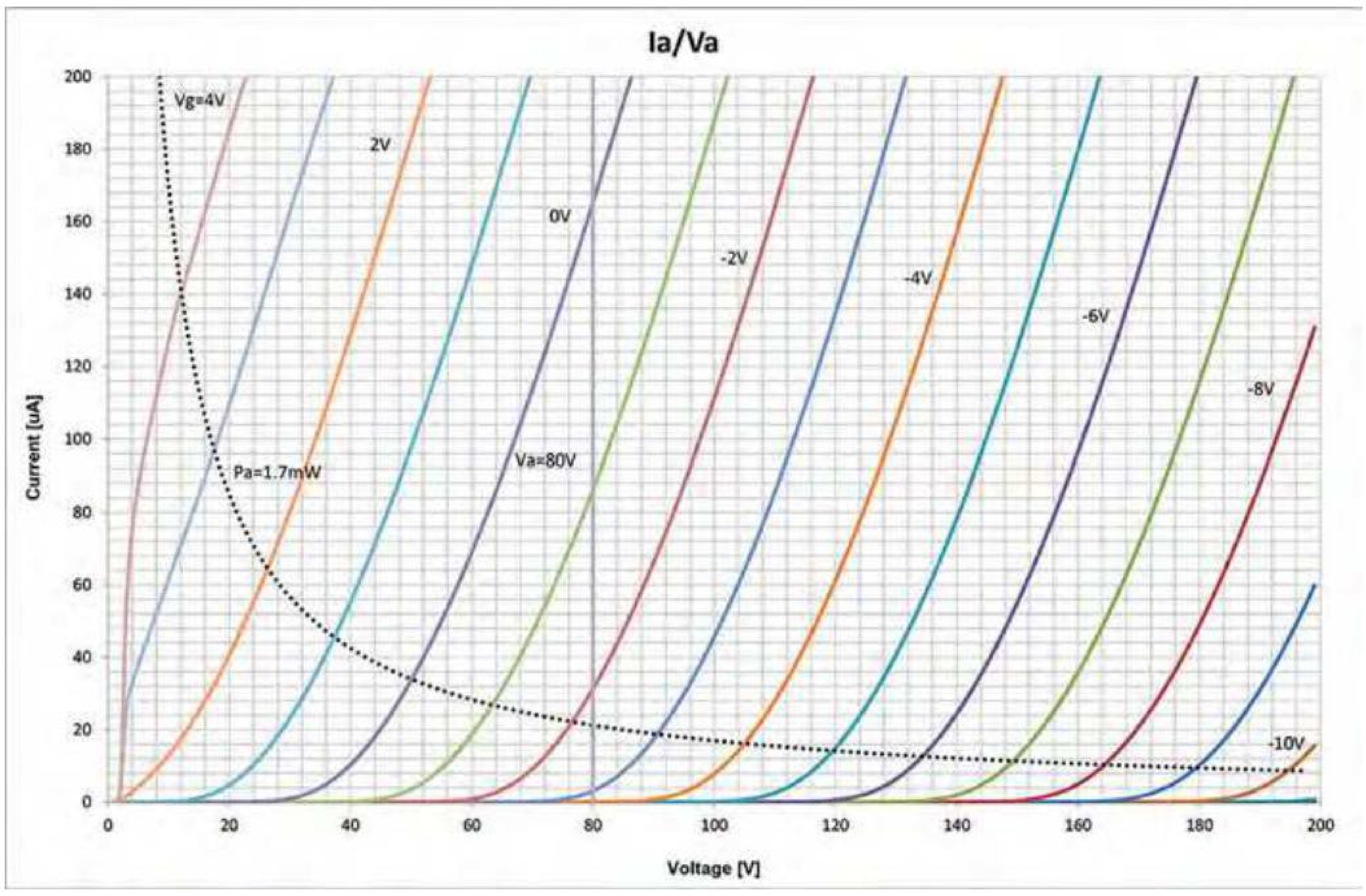


Fig.1: load lines for the Nutube triode showing the relationship between anode (plate) voltage (horizontal axis), anode/cathode current (vertical axis) and grid-cathode voltage (labels on curves). The area below the black dotted line is the continuous safe operation envelope.

The rise in distortion with frequency is mild, with THD+N only increasing by about 50% between 1kHz and 10kHz. The measurement shown in red on Fig.2 is with an ultrasonic (80kHz) bandwidth in order to measure the harmonics of higher test frequencies.

The blue trace gives a most realistic measurement up to about 10kHz, which then falls off due to the 22kHz filter limit cutting out the harmonics.

You may wish to compare Fig.2 and Fig.3 with Fig.12, which shows a spectral analysis of the distortion at 1kHz and around 200mV output.

As you can see from Fig.12, this method of reading the distortion gives much the same result as the Audio Precision system used to produce Fig.2 and Fig.3.

Fig.4 demonstrates that the preamp has a very flat response, with no peaks or wobbles. The output is down well under 1dB by 20Hz at the bass end, and an even smaller fraction of a decibel by 20kHz at the upper end. This plot has an extended frequency range of 10Hz-100kHz so you can get an idea of the actual –3dB points.

Fig.5 shows the channel separation. This is produced by feeding a signal into the right channel, monitoring the left channel output level and sweeping the test signal across the audible frequency range. The channels are then swapped, and the test is repeated.

As you can see, there is more coupling from the right channel to the left, and the separation figures are not amazing, at around 45-68dB.

However, this is more than good enough for a stereo system, and sounds panned entirely to the left or the right will still appear to be coming from just one speaker.

Fig.6 is a scope grab showing the preamp output (at the top, yellow) at around 200mV, 1kHz, with the ~0.07% residual distortion signal below, in blue. You can see that this is primarily third harmonic, with some second harmonic.

Fig.7 shows the much higher-level distortion present in the output if the triode is adjusted further away from its ideal operating point. This is around 0.3% THD+N, the majority of which is second-harmonic distortion.

Fig.8 shows the noise residual when the output level is set much lower. This is a fairly typical wideband white noise signal.

Circuit description

The full circuit is shown in Fig.9. One of the triodes in the Nutube provides amplification for the left channel (V1a), while the other triode is used for the right channel (V1b). These are connected as common-cathode amplifiers, where the cathode filament is referenced to ground. The signals are applied to the grids, and the resulting amplified signals appear at the corresponding anode (or plate).

The anode loads are 330kΩ resistors from the positive supply, with 150Ω/100µF low-pass filters to prevent supply noise from reaching the anodes.

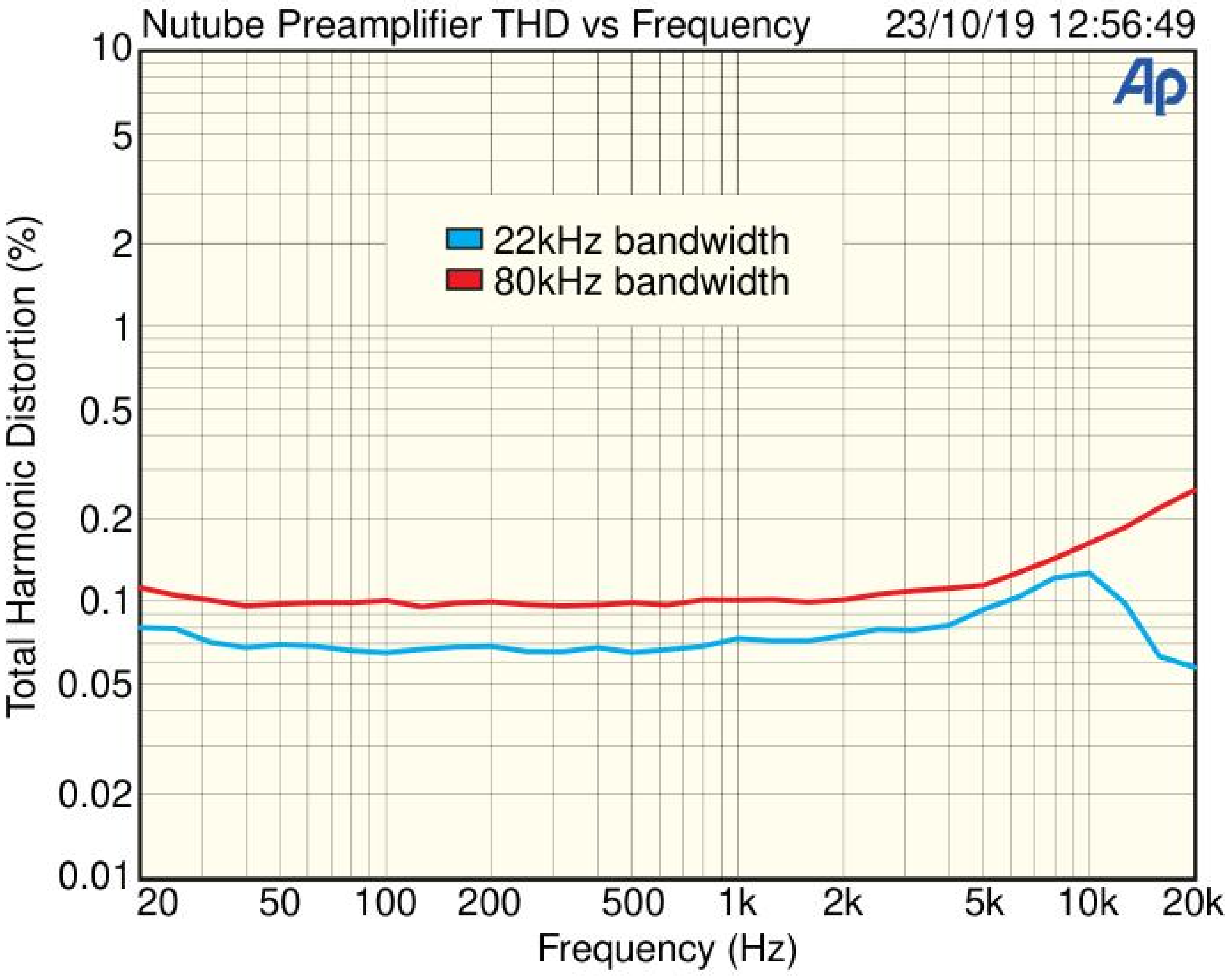


Fig.2: a plot of total harmonic distortion, including noise, against signal frequency. These measurements were made at about unity gain, with around 200mV RMS in/out, and with two different filter bandwidths. The blue curve (20Hz-22kHz) includes the distortion products and noise which are audible to the human ear, while the red curve (20Hz-80kHz) includes higher harmonics for more realistic readings at higher frequencies (8kHz+).

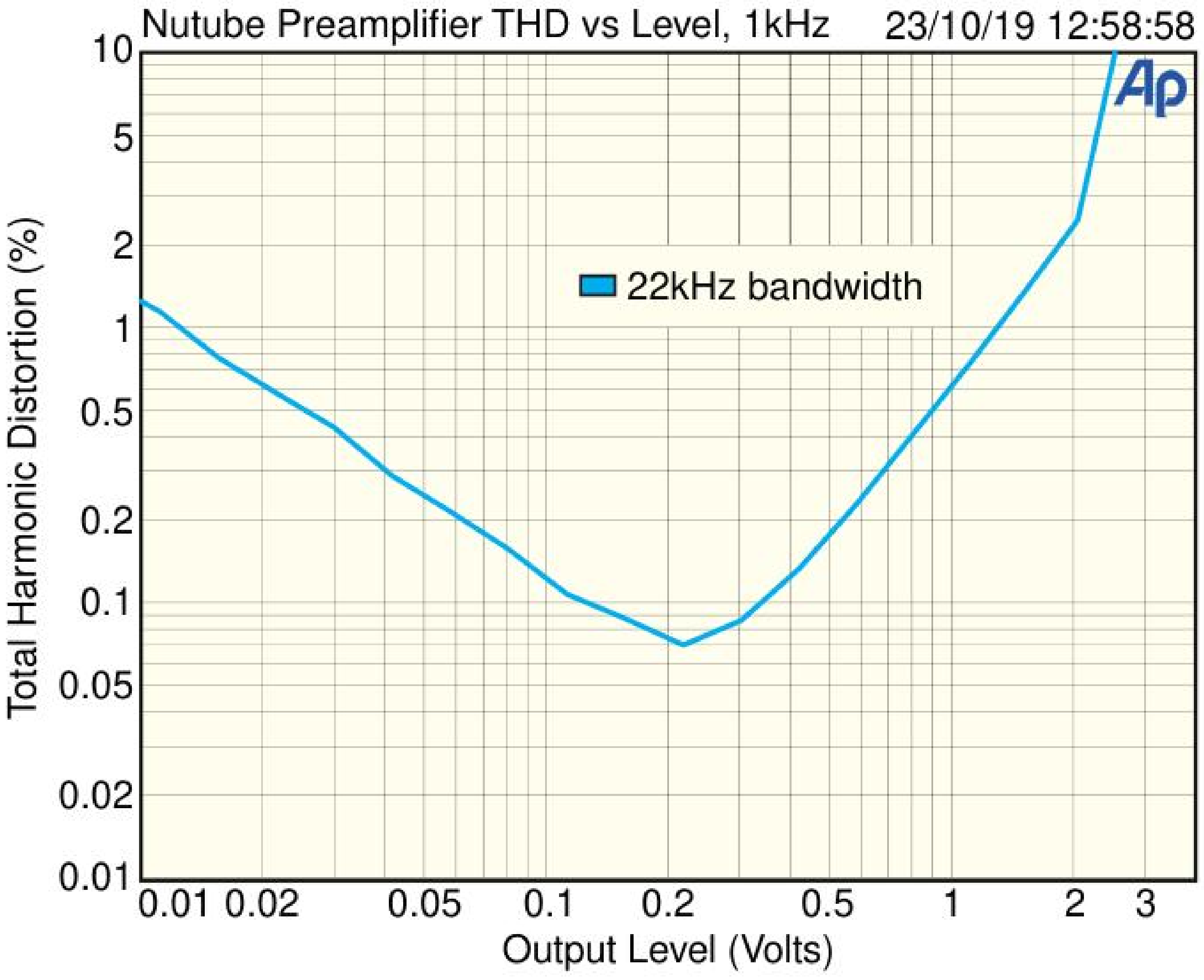


Fig.3: distortion plotted against output level. This graph demonstrates that the output level is the largest determining factor in the preamp's distortion performance. At low levels, noise begins to intrude, while at high levels, the waveform shape gets 'squashed' and so distortion increases significantly. The middle section, where distortion is lowest, is the range in which the preamp will generally be used.

The Nutube triodes have relatively low input impedances at the grids and high output impedances at the anodes, so op amp buffers are used at both ends. IC1a and IC2a ensure that the grids are driven from low impedances. IC1b and IC2b minimise the anode loading, as they have very high input impedances of 600MΩ, which is effectively in parallel with 1MΩ resistors.

These op amps have very low noise figures (3.3nV/√Hz) and distortion (0.00006% @ 1kHz and 3V RMS) when operated at unity gain. Therefore, these op amps do not affect the sound of the signals. The properties of the Nutube triodes dominate any effect that the op amps have on the signals.

We'll now describe the signal path in more detail, but only for the left channel, as both channels are almost identical. The input signal is fed in via RCA socket CON1a and passes through a 100Ω stopper resistor and ferrite bead (FB1). These, in conjunction with the 100pF capacitor, significantly attenuate RF signals entering the circuit, which could result in unwanted radio-frequency detection and reception.

The signal is AC-coupled to 50kΩ volume control VR1a via a 470nF DC-blocking capacitor. This capacitor removes any DC voltage that may be present at the input to prevent pot crackle, and also produces a low-frequency roll-off below about 7Hz. The signal is then AC-coupled from VR1a's wiper to the non-inverting input (pin 3) of op amp buffer IC1a via a 100nF capacitor.

Pin 3 of IC1a is biased near to half the supply voltage via a 1MΩ resistor that is tied to a half supply rail (Supply/2). The input bias current at pin 3 of IC1a will cause the DC voltage level to shift from this half supply level due to the current flowing through the 1MΩ resistor. This causes the signal voltage to rise about 0.5V above the half-supply rail, reducing the maximum symmetrical voltage swing.

But since the nominal supply voltage is 9V (down to 7.2V if the 9V battery is getting flat), the signal swing is still sufficient to prevent signal clipping of line-level audio signal levels.

IC1a's output drives V1a's grid (G1) via a 10µF coupling capacitor. This grid is DC-biased via a 33kΩ resistor with a voltage that's set using trimpot VR2. This is adjusted to set the operating point, and hence the distortion, produced by V1a.

V1a's plate anode load is a 330kΩ resistor which connects to either the Vaa or 6V supply via a 150Ω decoupling resistor. Which supply is used depends on the position of jumper JP1. When a 9V battery is used for power, using the fixed 6V selection prevents anode (plate) voltage variations as the battery discharges. When used with an external regulated supply, the Vaa setting would be selected.

The high-impedance amplified anode signal is again an AC-coupled op amp buffer, IC1b via a 100nF capacitor. IC1b is also biased to half supply via another 1MΩ resistor to Supply/2. This 1MΩ resistor loads the anode, reducing the Nutube anode signal to 75% of the unloaded signal. This is unavoidable in a circuit with such high impedances.

Note that the signal at the triode's anode is inverted compared to that applied to the grid. In some cases, it is important to maintain the phase of audio signals between the inputs and outputs. So the output signal from the triode is re-inverted by op amp IC3a, connected as an inverting amplifier.

VR4 is included so that the gain of IC3a can be adjusted. The gain of IC3b in the right channel is fixed at –2.3 times (–5.1kΩ ÷ 2.2kΩ), so the gain for IC3a is typically set at a similar level. The gain may need to be slightly different between the two channels to get equal gains for both outputs, due to variations in gain between the two triodes at similar bias levels.

Finally, the signal from IC3a is AC-coupled with a 10µF capacitor to remove the DC voltage and DC-biased to 0V with a 100kΩ resistor. The output is fed through a 150Ω isolation resistor to prevent oscillation of IC3a should long leads with a high total capacitance be connected.

To prevent noises when power is switched on and off, the output signal passes to the output RCA sockets via a pair of relay contacts that are open when power is off. At power-on, the relay is only switched on to allow signal through to the output terminals after everything has settled down. At power off, the relay is switched off immediately. This isolates the signal while the power supply voltages decay.

Filament current

Just like a traditional valve, the Nutubes have heater filaments. These are connected between F1 and F2 for V1a, and between F2 and F3 for V1b. So the F2 connection is shared between the two.

There are two ways to drive the filaments. One is to supply current to F1 and F3 via separate resistors and have the

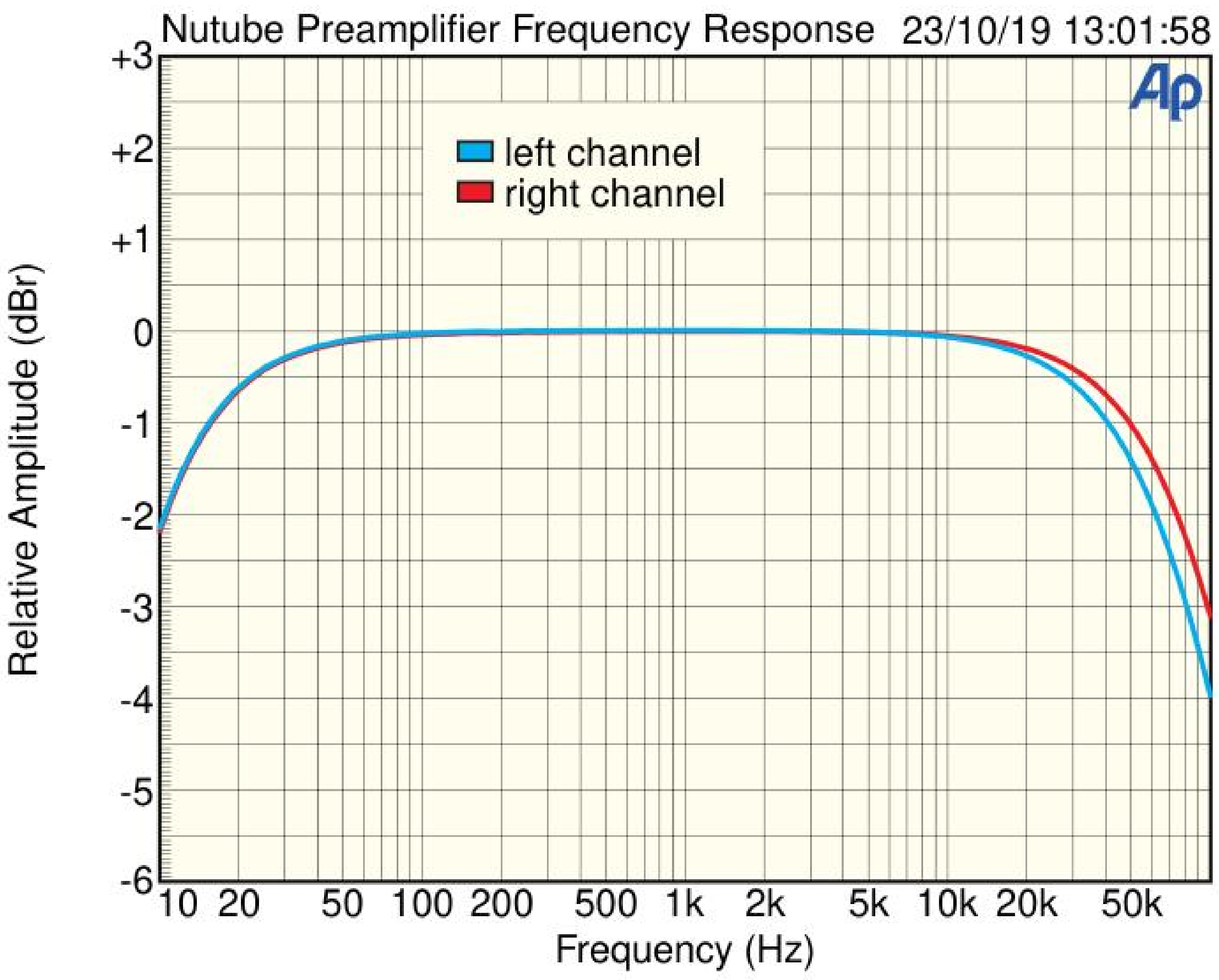


Fig.4: this plot shows that the *Nutube Preamplifier's* frequency response is commendably flat. This plot extends down to 10Hz and up to 100kHz so that you can see the roll-off at either end. The slight difference between the response of the two channels above 10kHz is likely due to slightly different biasing; we had purposefully biased the two channels slightly differently to see the difference in distortion.

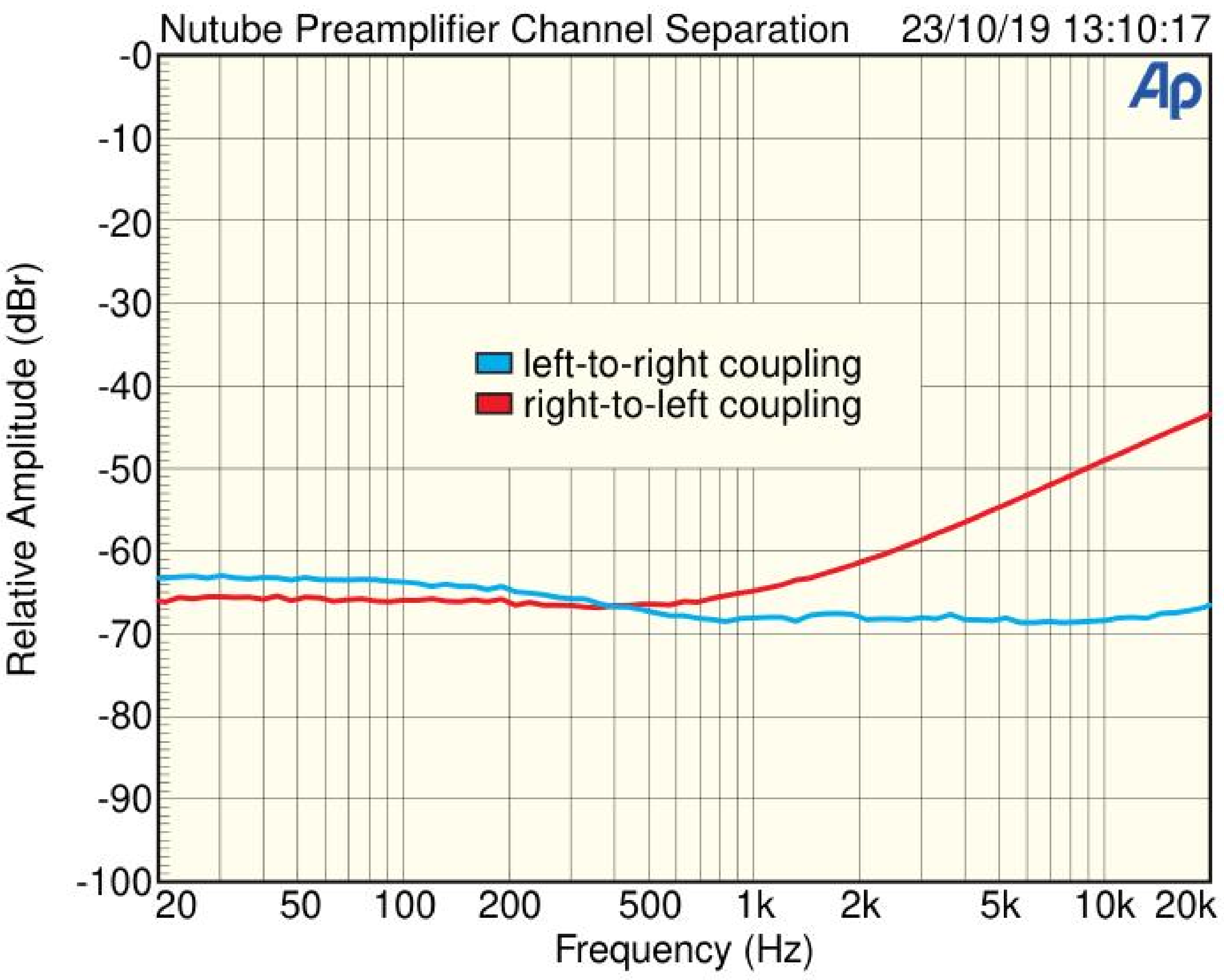


Fig.5: this graph shows the *Nutube Preamplifier's* channel separation. It is quite decent up to about 2kHz, with more than 60dB separation between channels. The main concern with signal coupling from one channel to another is that it introduces distortion; however, as this is not an ultra-low-distortion device, it isn't that big of a concern. We included this plot mostly for completeness.

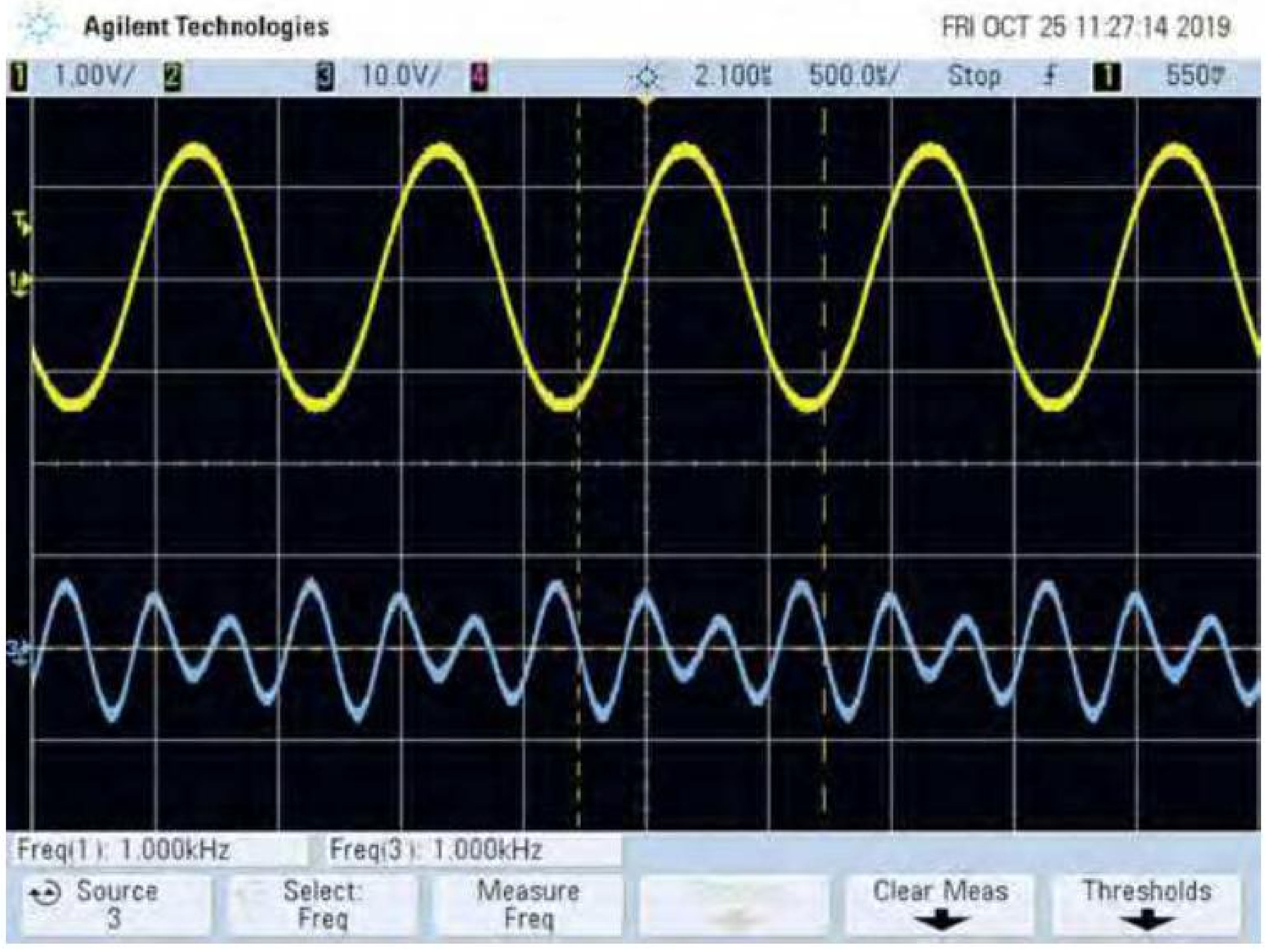

Agilent Technologies
FRI OCT 25 11:27:14 2019
1 1.00V/ 2 3 10.0V/ 4 2.100s 500.0s/ Stop 1 550mV
Freq(1): 1.000kHz
Freq(3): 1.000kHz
Source 3
Select: Freq
Measure Freq
Clear Meas
Thresholds

Fig.6: the output of the unit with the triode biasing adjusted for lowest distortion. The yellow trace is the output signal, while the blue trace is the distortion residual (ie, the yellow trace with its fundamental removed). It contains significant second and third harmonics.

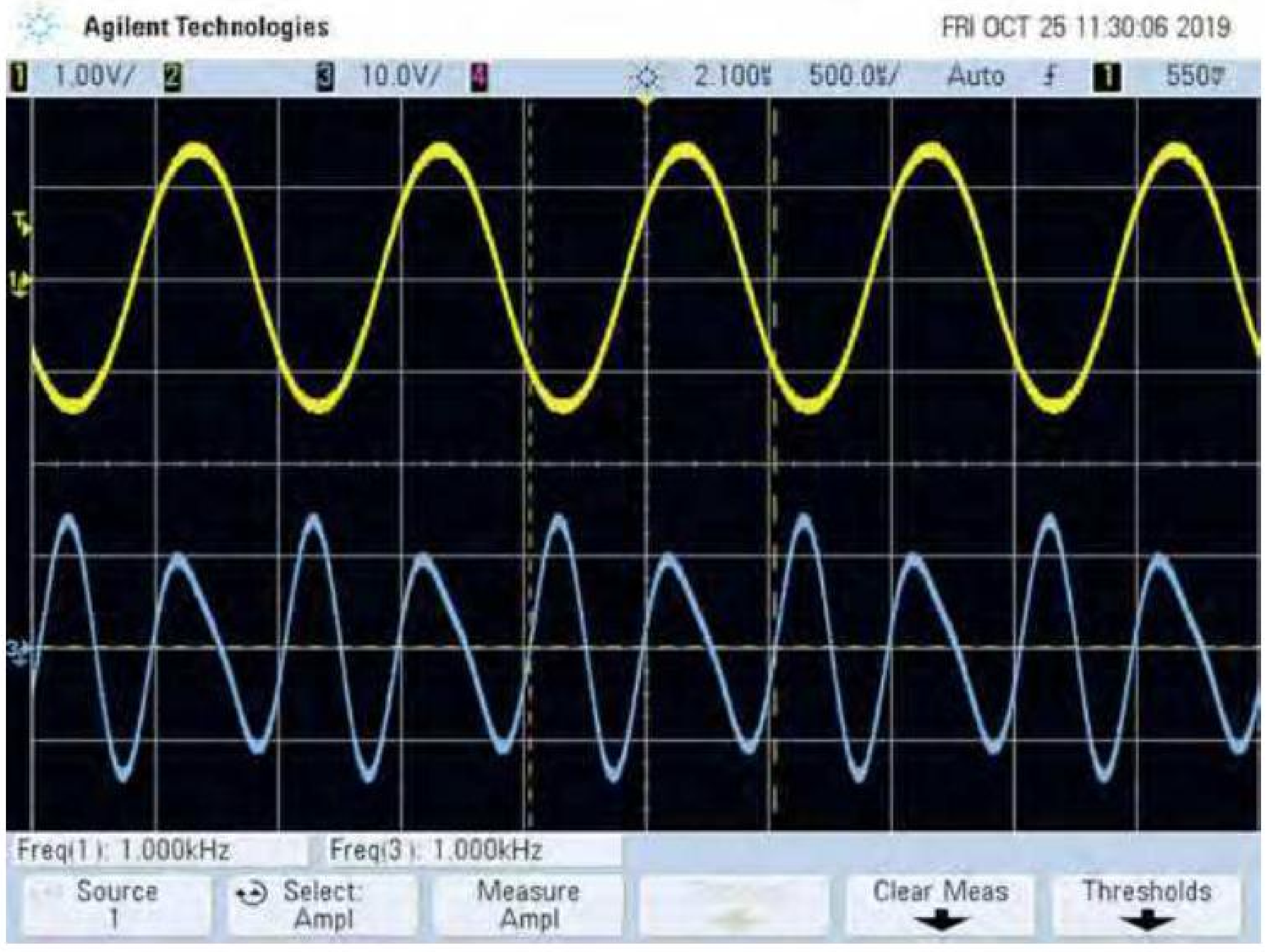

Agilent Technologies
FRI OCT 25 11:30:06 2019
1 1.00V/ 2 3 10.0V/ 4 2.100s 500.0s/ Auto 1 550mV
Freq(1): 1.000kHz
Freq(3): 1.000kHz
Source 1
Select: Ampl
Measure Ampl
Clear Meas
Thresholds

Fig.7: this plot is the same in Fig.6, but the triode biasing has been adjusted away from its optimal condition. Total harmonic distortion has risen to around 0.3%, with the second harmonic now the dominant distortion signal.

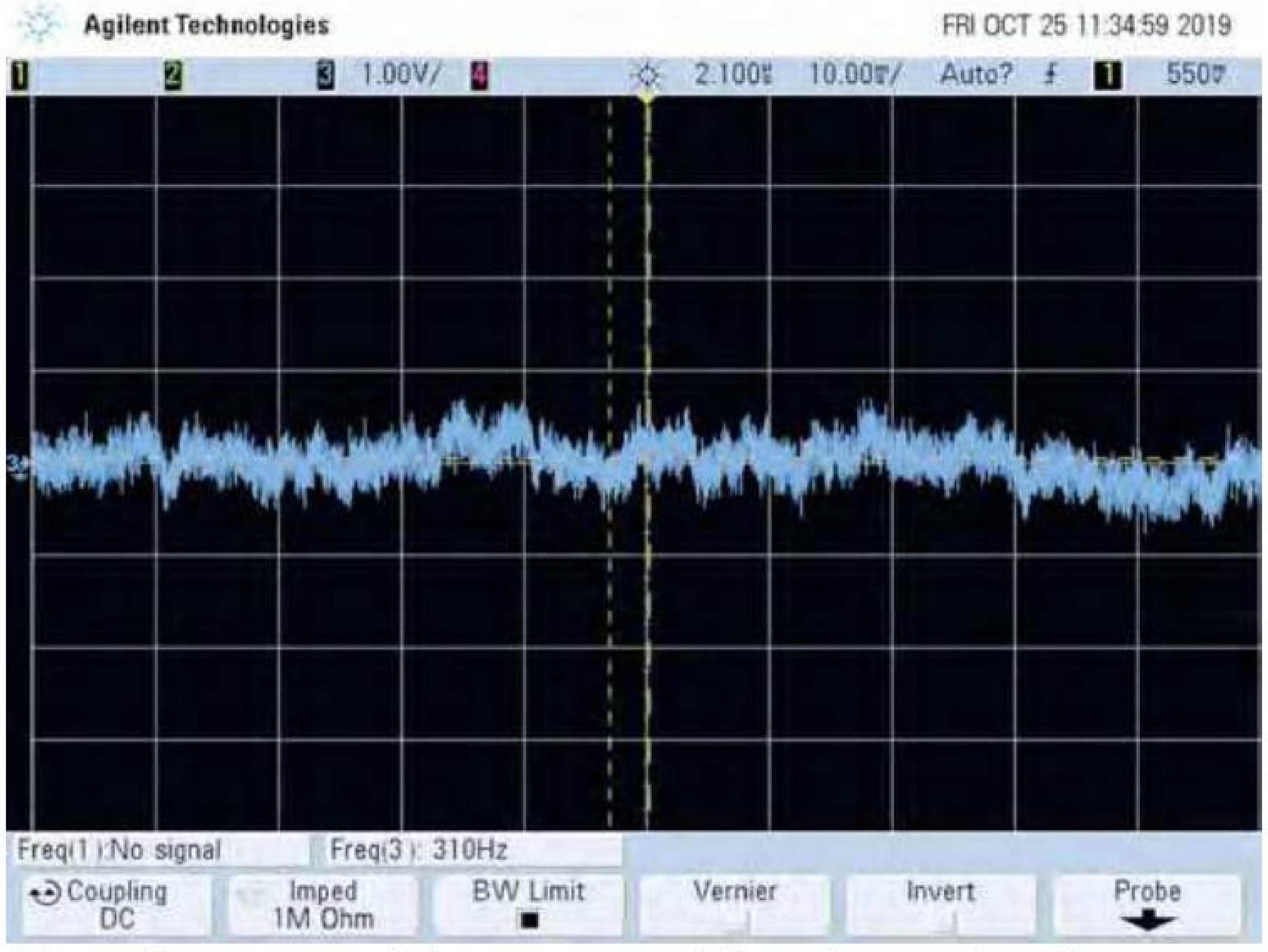

Agilent Technologies
FRI OCT 25 11:34:59 2019
1 2 3 1.00V/ 4 2.100s 10.00s/ Auto? 1 550mV
Freq(1):No signal
Freq(3): 310Hz
Coupling DC
Imped 1M Ohm
BW Limit
Vernier
Invert
Probe

Fig.8: the output of the preamp with no input signal. Some devices produce more high-frequency or more low-frequency noise. In this case, it appears quite close to white noise.

common F2 terminal tied to ground. In this case, the resistors are chosen for 17mA flowing in each filament, giving a total filament current of 34mA.

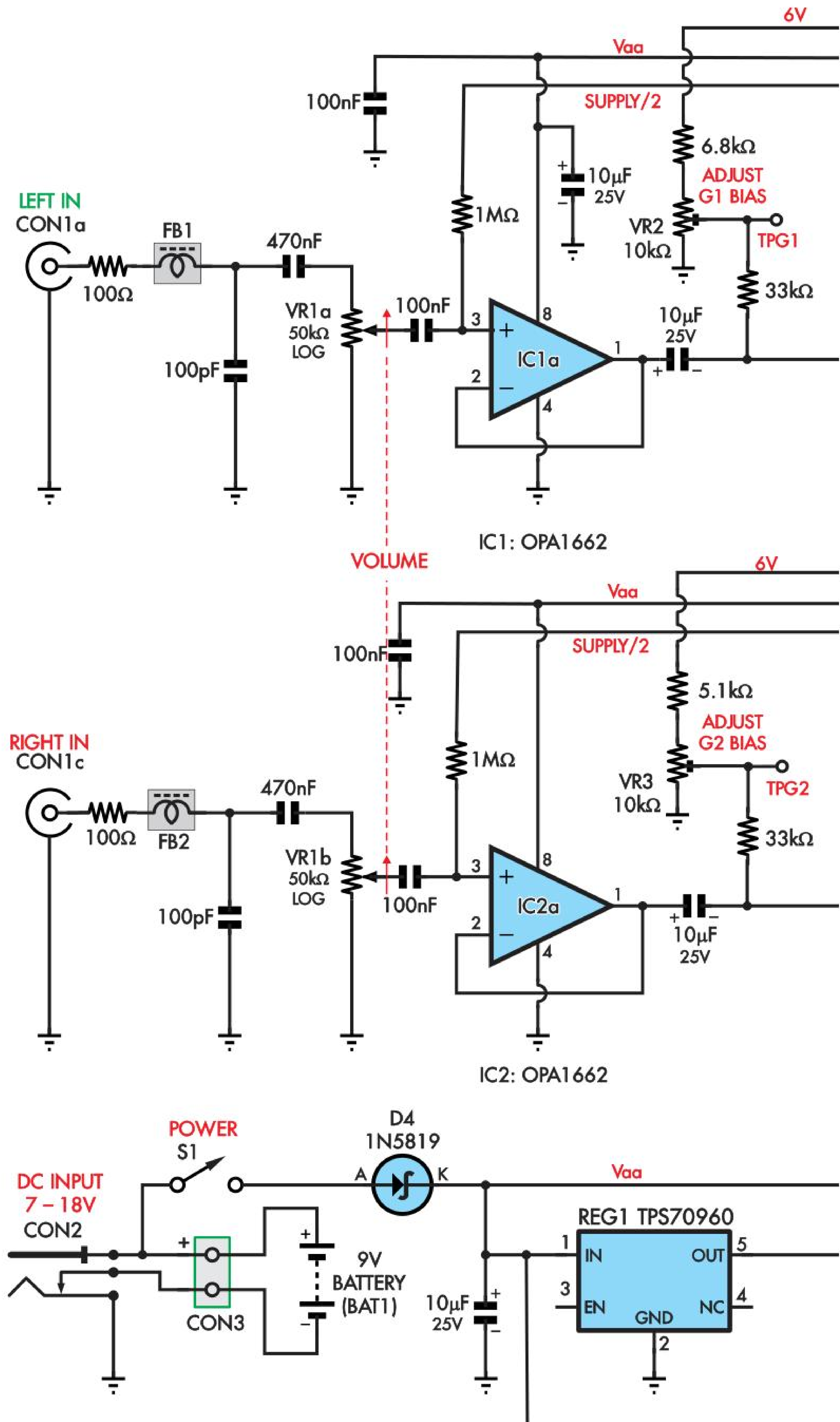
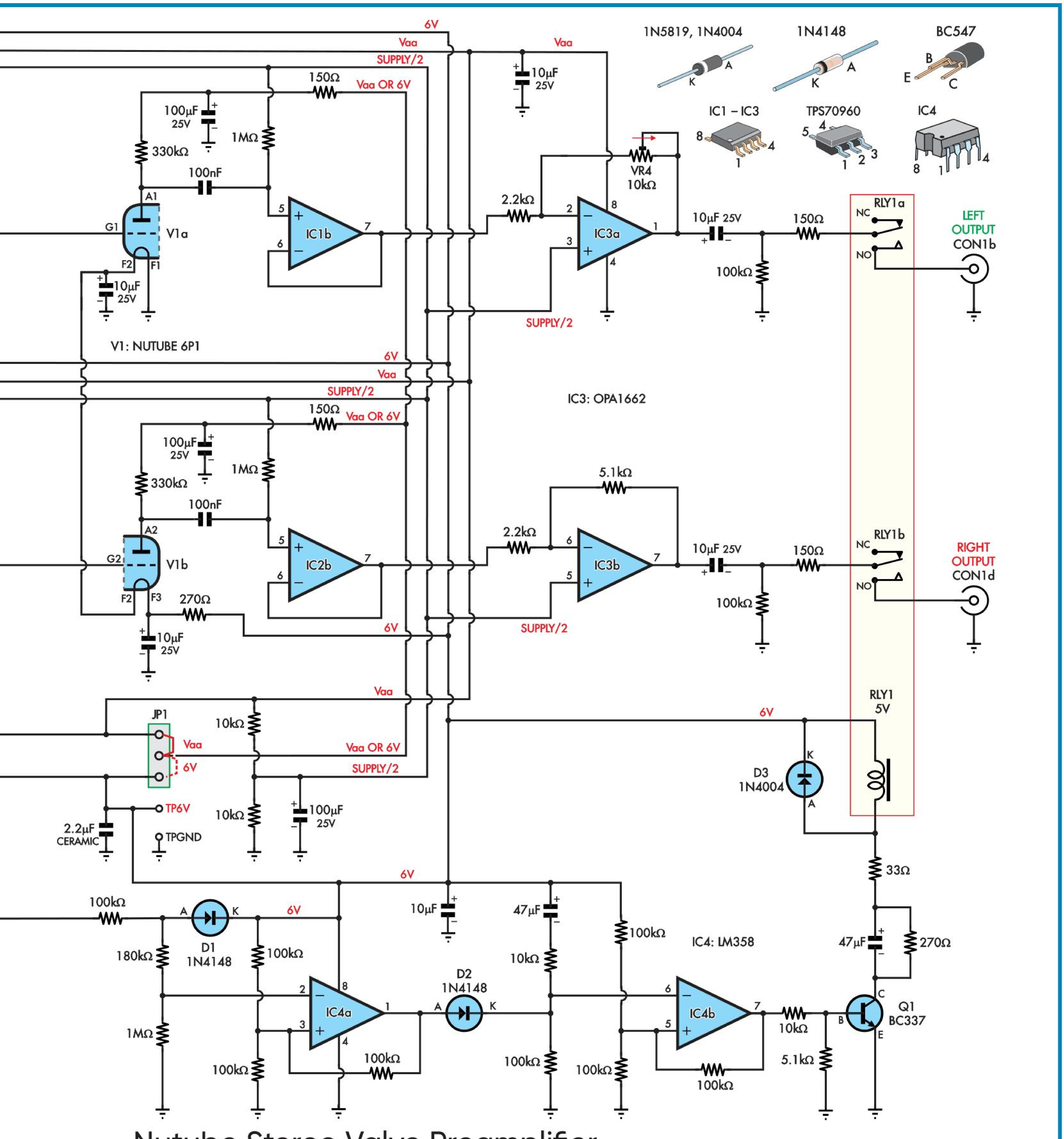

6V
Vaa
SUPPLY/2
100nF
10μF 25V
6.8kΩ
ADJUST G1 BIAS
VR2 10kΩ
TPG1
1MΩ
LEFT IN
CON1a
FB1
470nF
100Ω
VR1a 50kΩ LOG
100nF
100pF
33kΩ
10μF 25V
IC1a
IC1: OPA1662
VOLUME
6V
Vaa
SUPPLY/2
100nF
5.1kΩ
ADJUST G2 BIAS
VR3 10kΩ
TPG2
1MΩ
RIGHT IN
CON1c
100Ω
FB2
470nF
VR1b 50kΩ LOG
100nF
100pF
33kΩ
IC2a
10μF 25V
IC2: OPA1662
POWER S1
D4 1N5819
A K
Vaa
DC INPUT 7 – 18V
CON2
9V BATTERY (BAT1)
CON3
10μF 25V
REG1 TPS70960
IN OUT EN NC GND
1 5 3 4 2

Fig.9: the input signals from CON1a and CON1c pass through RF filters and volume control pot VR1 before being AC-coupled to ultra-low-distortion buffer op amps IC1a and IC2a. These feed the signals to the grids of V1a and V1b, while VR2 and VR3 allow you to adjust the DC grid bias levels. The inverted output signals at the anodes of V1a and V1b are AC-coupled to the inputs of buffer op amps IC1b and IC2b. The signals are then re-inverted by op amps IC3a and IC3b before being fed to the outputs via the contacts of RLY1. VR4 allows the gain of the two channels to be matched. IC4 controls RLY1's coil so that it switches on around five seconds after power is applied, and switches off immediately upon power removal, eliminating clicks and thumps.

But in our circuit, we connect the filaments in series, so the same 17mA flows through each filament for a 17mA total current, but with twice the voltage across the filaments. This is a more efficient way to drive the filaments, and saves power when using batteries.

In our circuit, F1 is tied to ground, F2 is left open and current supplied via a 270Ω resistor from 6V to F3 ((6V – 0.7 – 0.7) ÷ 270Ω = 17mA). Note that F2 and F3 are bypassed to ground with 10μF capacitors. This reduces noise in the circuit.

There is one extra consideration when the filaments are in series. As the Nutubes are directly heated, V1b's cathode will be 0.7V higher than V1a, due to the voltage drop across V2's filament before the current reaches V1. This changes the


6V
Vaa
SUPPLY/2
150Ω
Vaa OR 6V
10µF 25V
100µF 25V
1MΩ
330kΩ
100nF
A1
G1
V1a
F2
F1
IC1b
VR4 10kΩ
2.2kΩ
IC3a
10µF 25V
150Ω
100kΩ
RLY1a
NC
NO
LEFT OUTPUT
CON1b
V1: NUTUBE 6P1
IC3: OPA1662
1N5819, 1N4004
1N4148
BC547
IC1 – IC3
TPS70960
IC4
A2
G2
V1b
F3
270Ω
IC2b
5.1kΩ
IC3b
RLY1b
RIGHT OUTPUT
CON1d
JP1
10kΩ
TP6V
TPGND
2.2µF CERAMIC
RLY1 5V
D3 1N4004
33Ω
47µF
270Ω
100kΩ
D1 1N4148
180kΩ
1MΩ
IC4a
D2 1N4148
10µF
47µF
IC4: LM358
IC4b
Q1 BC337
5.1kΩ

Nutube Stereo Valve Preamplifier

bias voltage requirement at the grid (G2) for V1b compared to G1 for V1a. The extra voltage required for G2 is provided by having a wider voltage range for VR3 due to a lower-value resistor connecting it to the 6V supply compared to VR2.

Note that the grid bias voltage derived from VR2 and VR3 is relative to the output of 6V regulator REG1. This is a fixed voltage, so the grid bias voltage does not vary with the supply voltage.

Power supply

When no DC plug is inserted into DC socket CON2, the internal 9V battery supplies power to the circuit, via CON2's normally closed switch connecting the negative of the battery to ground. When a power plug is inserted, power is from the DC input and the battery negative is disconnected.

Power switch S1 connects power to the rest of the circuit, whether from the battery or an external source, while diode D4 provides reverse-polarity protection.

REG1 is a low-dropout, low-quiescent-current 6V regulator. It is included to maintain a constant grid voltage for the Nutube when power is from a battery, as battery voltage naturally varies over time. The 6V rail also powers relay RLY1.

The input of REG1 is bypassed with a 10µF capacitor, while a 2.2µF ceramic capacitor filters the output. This output capacitor has the required low ESR (effective series resistance) to ensure stability at the regulator output.

The half-supply rail is derived from a pair of 10kΩ resistors connected in series across the anode supply for V1. It is bypassed with a 100µF capacitor to reduce noise and lower the rail impedance.

Power switching and output isolation

As mentioned earlier, the relay contacts at the left and right outputs connect the signals some time after power-up and disconnect the signals quickly when power is switched off. IC4, Q1, RLY1 and associated components provide this signal switching.

IC4a and IC4b are two halves of an LM358 single-supply, low-power dual op amp. They are used as comparators with hysteresis. The hysteresis is provided by 100kΩ resistors from their outputs to their non-inverting inputs, while the nominal comparator threshold at these inputs is set around 2V when the output is low and 4V when the output is high.

So in each case, the output goes high when the voltage at the inverting input drops below 2V, and then goes low again when the voltage at the inverting input rises above about 3.5V (you might expect 4V, but the LM358's output can't swing to the positive rail). In other words, there is about 1.5V of hysteresis.

RLY1 is initially off, and when power is applied via switch S1, several things happen. First, power is supplied via D1 to the preamplifier circuitry, including REG1, V1 and IC1-IC4. The supply and signal-coupling capacitors begin to charge up to their operating conditions.

At the same time, the inverting pin 2 input to IC4a is pulled high, to near the incoming supply voltage, via the 100kΩ and 180kΩ resistors connecting to switch S1. Diode D1 prevents more than 6.5V from being applied to this pin.

The 180kΩ and 1MΩ resistors form a voltage divider so that their junction tends to sit at around 5.5V when there is more than 6.5V at the anode of D4.

This is above the pin 3 voltage, and therefore the output of IC4a goes low, near 0V. Pin 3 is therefore around 2V. Diode D2 is reverse-biased and pin 6, the inverting input of IC4b, is initially held high near to 6V, due to the 47µF capacitor being initially discharged. The 10kΩ resistor in series with the capacitor reduces the pin 6 voltage down to about 5.7V initially.

This is above the 4V at the non-inverting pin 5 input, so the output of IC4b will be low. Pin 5 will be at 2V. The low output of IC4b means NPN transistor Q1 is off, and the relay is off. The relay contacts will be open, so no audio passes through to the output.

As the 47µF capacitor charges via the 10kΩ and 100kΩ resistors, after about five seconds, the voltage at pin 6 will drop below the voltage at the pin 5 input (2V). The output of IC4b then goes high, driving transistor Q1 and switching on RLY1. The audio signals are then connected to the left and right-channel output sockets.

Note the 47µF capacitor with a parallel 270Ω resistor and series 33Ω resistor between the collector of Q1 and the coil of RLY1. The 33Ω resistor is included so that the 5V-rated relay coil is initially driven with 5V rather than the full 6V of the supply.

Then, as the 47µF capacitor charges, the voltage to the relay coil is reduced until it is instead supplied current via the 270Ω resistor. This reduces relay coil voltage and current, saving power but still holding the relay's contacts closed.

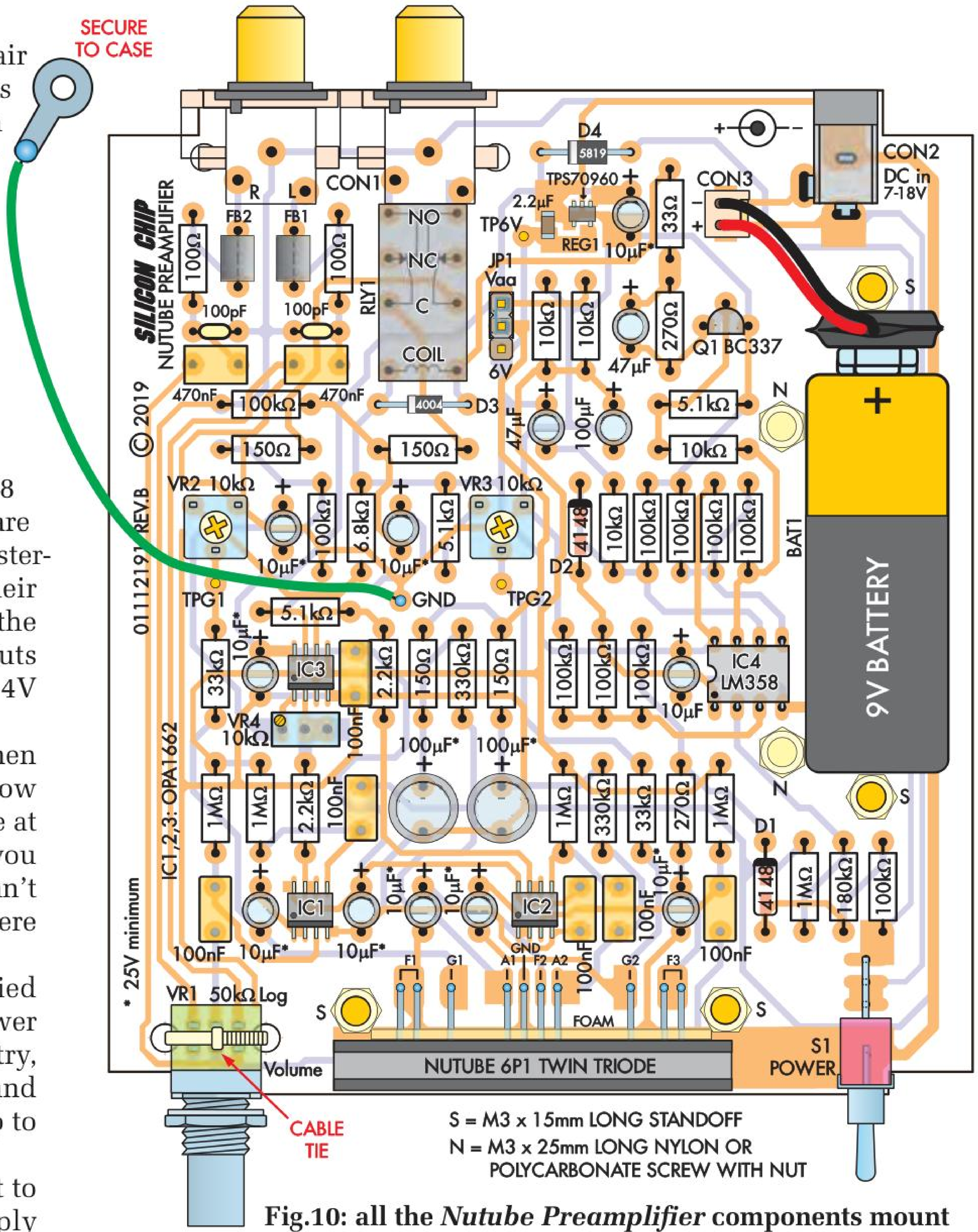

SECURE TO CASE
CON1
R
L
FB2
FB1
100Ω
NO
NC
C
COIL
RLY1
100pF
470nF
100kΩ
150Ω
D4
5819
TPS70960
2.2µF
TP6V
REG1
10µF*
33Ω
CON3
CON2
DC in
7-18V
JP1
Vaa
6V
10kΩ
47µF
270Ω
Q1 BC337
4004
D3
100µF
5.1kΩ
S
N
SILICON CHIP
NUTUBE PREAMPLIFIER
© 2019
01112191 REV.B
VR2 10kΩ
6.8kΩ
VR3 10kΩ
4148
D2
BAT1
9V BATTERY
TPG1
GND
TPG2
33kΩ
IC3
2.2kΩ
330kΩ
IC4
LM358
VR4
10kΩ
100nF
100µF*
1MΩ
IC1,2,3: OPA1662
D1
180kΩ
IC1
IC2
* 25V minimum
F1
G1
A1
F2
A2
G2
F3
FOAM
VR1 50kΩ Log
Volume
NUTUBE 6P1 TWIN TRIODE
S1
POWER
CABLE TIE
S = M3 x 15mm LONG STANDOFF
N = M3 x 25mm LONG NYLON OR POLYCARBONATE SCREW WITH NUT

Fig.10: all the *Nutube Preamplifier* components mount on one double-sided PCB, as shown here. They are mostly standard parts, but IC1-IC3 and REG1 are only available in SMD packages. The Nutube (V1) is in a SIL-type package with right-angle leads that are surface-mounted to pads on the top of the board. The whole assembly slides into an extruded aluminium case.

The value of the 270Ω resistor means that the current drawn by the relay coil drops from 30mA initially down to about 12.8mA, extending battery life.

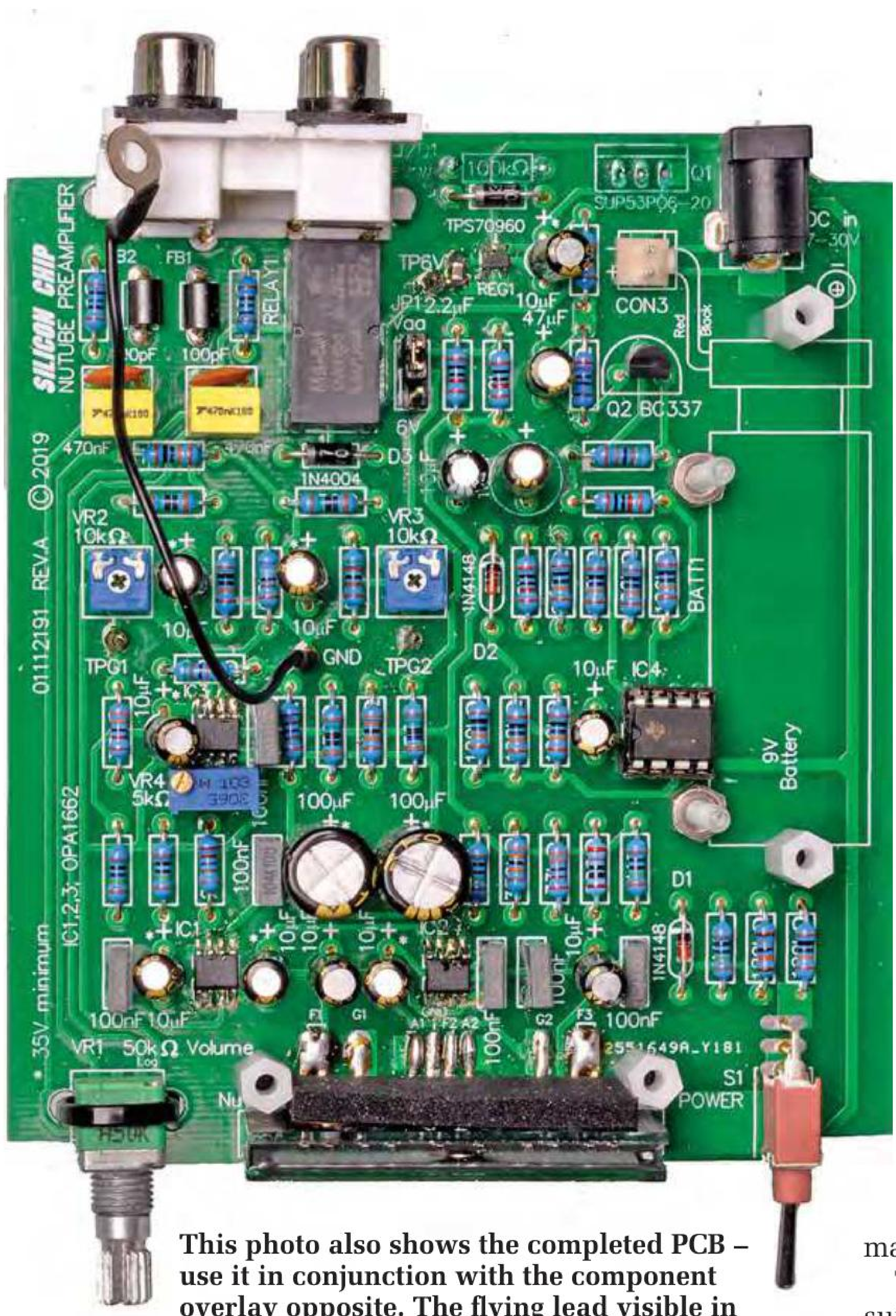
When power is switched off via S1, the pin 2 voltage at IC4a's input immediately drops to 0V. That voltage is below the pin 3 voltage, so IC4a's output goes high. Diode D2 conducts and pulls pin 6 of IC4b above the pin 5 threshold, so IC4b's output immediately goes low. Q1 switches off and the relay contacts open. This all happens well before the supply capacitors in the circuit have time to drop significantly in voltage. So the output signals are cut before anything in the circuit can misbehave.

The 10kΩ resistor between the diode D2 and the 47µF capacitor is there so that the pin 6 input to IC4b can be immediately taken high, without having to wait for the 47µF capacitor to discharge.

Construction

The *Nutube Stereo Preamplifier* is built using a double-sided PCB coded 01112191, which measures 98 × 114mm and is available from the *PE PCB Service*. It is housed in an extruded aluminium enclosure with clear end panels, measuring 115 × 51 × 119mm. Fig.10 has the PCB assembly details.

Start by fitting the surface-mount parts. Mostly, these are used because the same parts are not available in through-hole packages. They are not difficult to solder using a fine-tipped soldering iron.



This photo also shows the completed PCB – use it in conjunction with the component overlay opposite. The flying lead visible in this photo (and the photos below) earths the aluminium case to the PCB to minimise hum.

Good close-up vision is necessary, so you may need to use a magnifying lens or glasses to see well enough. These parts are IC1, IC2 and IC3, REG1 and its associated 2.2µF ceramic capacitor.

Make sure that each component is oriented correctly before soldering it – ie, rotated as shown in Fig.10. The ceramic capacitor is not polarised.

For each device, solder one pad first, check alignment and readjust the component positioning by reheating the solder joint if necessary before soldering the remaining pins.

If any of the pins become shorted with solder, solder wick can be used to remove the solder bridge. But note that pins 1 and 2, and pins 6 and 7 of both IC1 and IC2 connect together on the PCB, so a solder bridge between these pins is acceptable.

Continue construction by installing the resistors (use your DMM to check the values), followed by the two ferrite beads. Each bead is installed by using an offcut length of wire (from the resistors) feeding the wire through it and then bending the leads down through 90° on either side to fit the PCB. Push each bead all the way down so that it sits flush against the PCB before soldering its leads.

Install diodes D1-D4 next. Take care to orient each correctly, as shown in the overlay diagram, and make sure each is in its correct position (ie, don't get the different types mixed up) before soldering.

Following this, fit the IC socket for IC4. Make sure that the socket is seated flush against the PCB and that it is oriented correctly. It's best to solder two diagonally opposite pins of the socket first and then check that it sits flush with the board before soldering the remaining pins.

You could skip the socket and solder IC4 straight to the board. This would improve long-term reliability but would make it much more difficult to swap or replace IC4 should that be necessary.

The MKT and the two 100pF ceramic capacitors can now go in, followed by the electrolytic capacitors. The polarised electrolytics must be oriented with the correct polarity; ie, with the longer lead into the pad marked with the '+' sign.

Now install the two single-turn trim pots, VR2 and VR3. These might be marked as '103' rather than '10kΩ'.

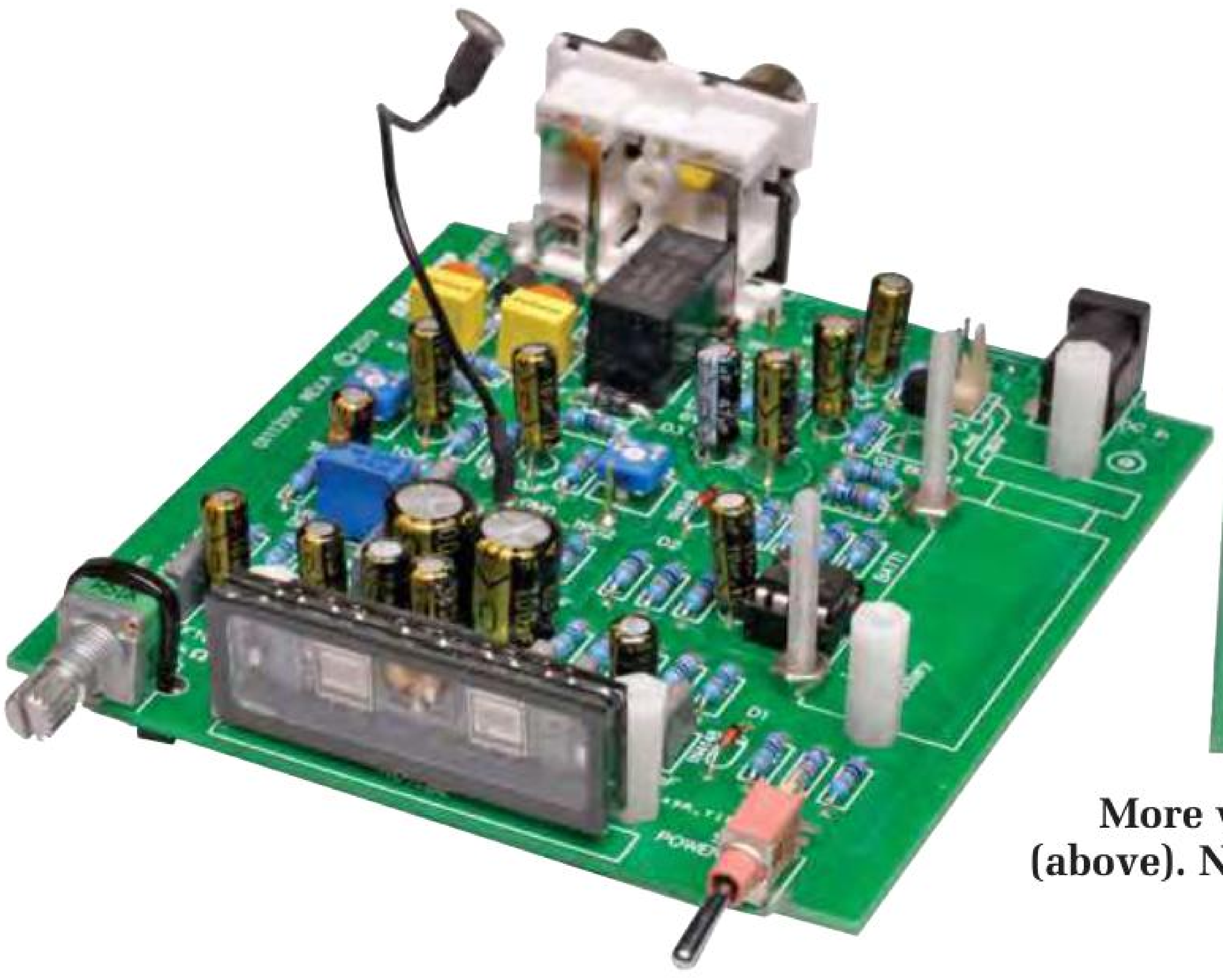
Next, mount multi-turn trimpot VR4. Orient it with the adjusting screw positioned to the left, as shown. It also may be marked as '103' instead of '10kΩ'.

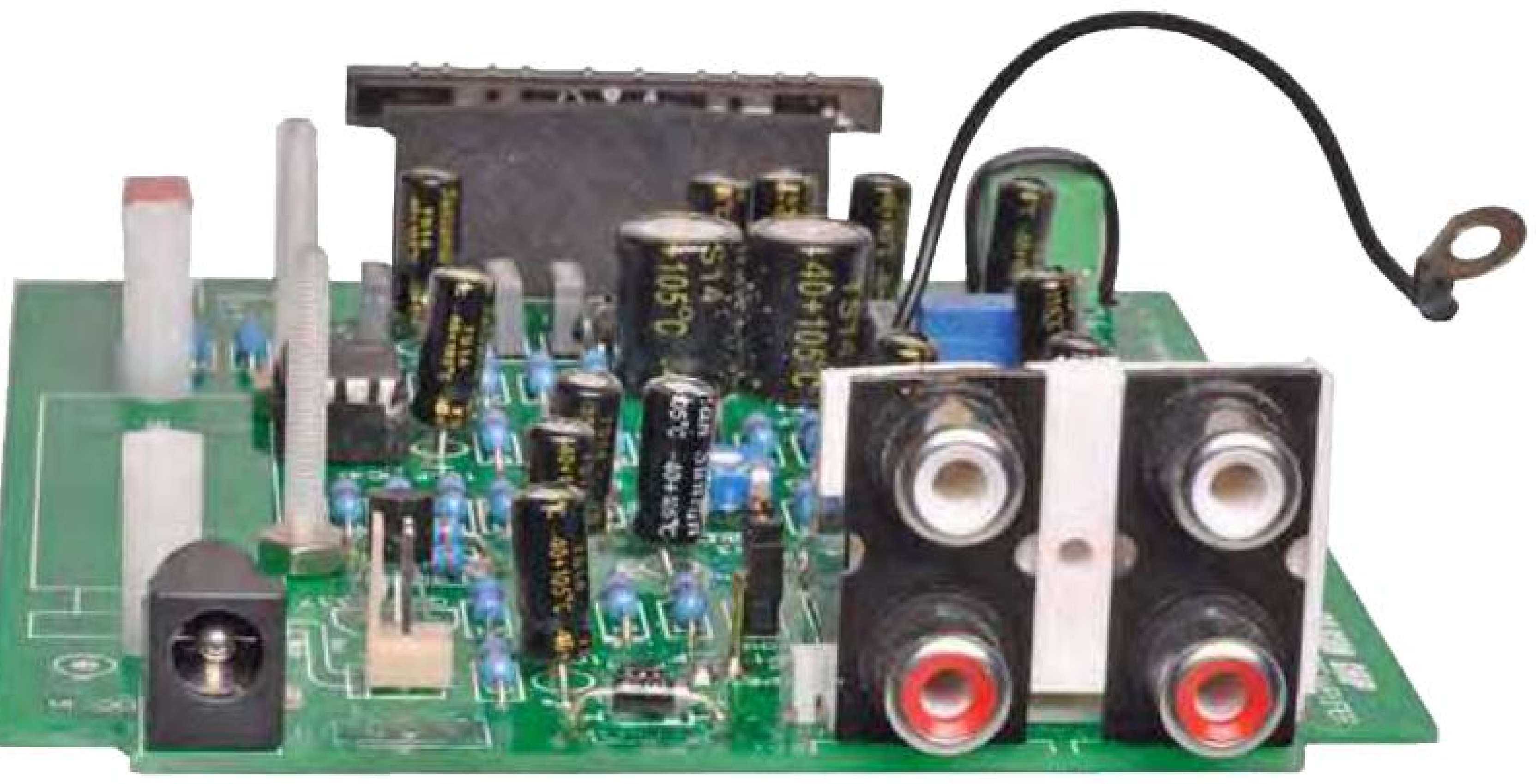
The next step is to fit Q1 by splaying its leads slightly to suit the hole arrangement on the PCB. Also install PC stakes for GND, TPG1, TPG2 and TP6V. The three-way header for JP1 and the two-way header for the battery lead can be mounted now, followed by RLY1, CON1, CON2 and switch S1.

Potentiometer VR1 is mounted, soldered in place and secured against the PCB using a cable tie around the pot body. This stops force on the shaft from breaking the solder joints or lifting tracks off the board.

Feed the tie through the holes in the PCB on each side of the pot, and tie it underneath.

Nutube V1 is mounted so that the front glass is vertical and with its leads soldered to the top pads on the PCB, similar to a surface-mount component. Pins F1 and F3 at each end of the Nutube use two adjacent leads on the Nutube device. In addition to the leads, it is supported by two 15mm-long tapped spacers, one on either side of the device, which hold a piece of foam against the Nutube envelope.





More views of the completed PCB from the front (at left) and the rear (above). Neither photo has the 9V battery in place, but its support standoffs and screws are ready for it.

Where can you buy a 6P1 Dual Triode?

The 6P1 is available from RS Components (**https://uk.rs-online.com/web/**). We have to warn you, though, it's not a cheap device: RS Components list it as £50 (inc VAT, plus postage – the RS stock number is 144-9016).

We would expect prices will eventually come down as they become more popular and more suppliers carry them.

The completed PCB simply slides into the extruded case so that the pot shaft and switch emerge from the front panel. No PCB screws are necessary as it is held tight by the front and rear case ends.

Secure these spacers to the PCB using short machine screws fed in from the underside of the PCB.

We will later sandwich the foam between the spacers and the Nutube, stopping it from flexing its leads too much. Also fit one 15mm standoff at each end of the battery outline on the PCB (see photos).

The sides of the battery are held in by two M3 x 25mm nylon or polycarbonate screws passed up from the underside of the PCB and secured with M3 nuts.

Wiring

Crimp and/or solder the battery wires to the header socket terminals after cutting these wires 60mm long. Then insert these terminals into the header socket shell, making sure you get the red and black wires in the correct positions, as marked on the PCB.

An earth wire is also required to prevent hum injection to the circuit if the case is touched. This connects the metal case to the GND terminal on the board. Solder it to the solder lug at one end and the GND terminal on the board at the other. Heatshrink tubing can be used over the lug terminal and PC stake for GND.

When the case is assembled, the solder lug is captured in the top corner end-cap screw, adjacent to the RCA terminals.

Powering up and testing

If you are planning to use a battery to supply power, connect a jumper shunt in the 6V position for JP1. That way, any voltage changes from the battery will not affect the anode plate voltage. If using a DC plugpack, use the Vaa position for JP1.

Initially, set VR2 and VR3 to midway. Apply power to the circuit from a 7-18V DC supply. Check that TP6V is between 5.88 and 6.12V. Also check the relay switches on after about five seconds; you should hear it click in.

Adjust VR2 so that the left-hand plate of the Nutube lights up at its brightest. Similarly, adjust VR3 so that the right-hand plate of the Nutube glows brightest. If using a supply that's over 12V, then make sure the grid voltage is less than 2.5V; otherwise, the device's maximum dissipation rating will be exceeded. The grid voltage for each triode can be measured at TPG1 and TPG2, relative to the GND PC stake.

VR4 adjusts the output of the left channel so that it can match the right channel in level. This can be done by connecting up the preamplifier to your sound system and rotating VR4 so both channels have the same output level, just by listening.

For more accurate adjustments, you need a signal generator. You can use a standard hardware-based signal generator, or computer software.

You will also need suitable leads to connect the generator to the RCA inputs. For connection to a computer, you typically need a stereo lead with RCA plugs one end and a stereo 3.5mm jack plug at the other. Leads for a hardware signal generator will require an RCA plug one end and a connector for the generator, such as a BNC plug, at the other.

Apply a 1kHz signal of about 1V RMS to the right channel preamplifier input (red input socket). Monitor the right channel output with a multimeter set to measure AC volts.

Set the volume control for about 500mV signal at the output. Adjust VR3 for maximum signal, but when doing this, adjust the volume control so the level does not exceed about 500mV. That's required to ensure the signal is not clipped. When the maximum level is found, take note of the level reading.

Now apply the same signal to the left channel (white RCA input) and measure the left channel output. Do not change the volume setting, but you may need to adjust VR4 for a suitable level, not much more than 500mV. Adjust VR2 for maximum signal as before. Now adjust VR4 so that the measured level is the same as that already measured in the right channel.

If you wish to set the grid bias more accurately, spectrum analyser software can be used. The spectrum analyser will show the distortion products of the preamplifier, including the fundamental and harmonics. The fundamental is the reproduction of the actual applied signal.

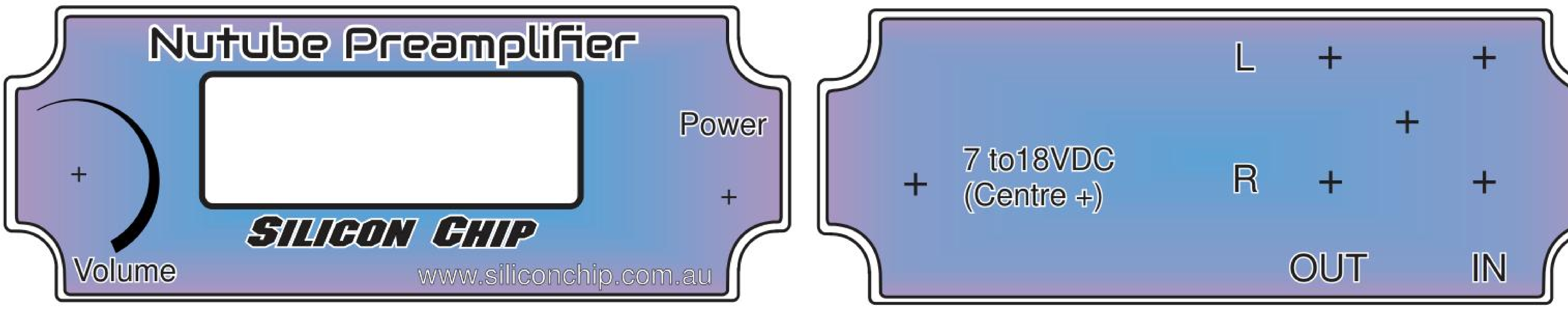


Fig.14: the 1:1 front and rear panel artwork can also be used as a template. V1 requires a 43 × 15mm cutout; the volume control, a 10mm hole; and the power switch, a 5mm hole. On the rear panel, the RCA sockets require 10mm holes where shown, with a 3mm hole in the middle; the DC socket is 5mm. (Download from the January 2021 page of the *PE* website.)

With a perfect preamplifier, without distortion, you would only see the fundamental at the output.

However, with a real preamplifier, there will be noise and distortion. This will show up in the analyser as other spikes rising above the noise floor.

Typically, the distortion will have second, third, fourth, fifth harmonics and so on. For a 1kHz signal, the fundamental (first harmonic) would show as a peak at 1kHz, with the second harmonic at 2kHz, the third harmonic at 3kHz, the fourth at 4kHz... These harmonic distortion products hopefully will be at a lower level than the fundamental, and not all harmonics will necessarily be present.

Once you can see this, you can adjust the grid bias for minimum distortion. For that matter, you could also adjust it for maximum distortion, if that's what you're after! (See panel on the next page).

Final assembly

The *Nutube Preamplifier* PCB is housed inside an aluminium enclosure with clear end panels, measuring 115 × 51 × 119mm.

If you are not using a battery for power, unplug the battery clip from CON3 to prevent the contacts from shorting onto a part of the circuit.

The end panels include 3mm-thick foam plastic that can be used as padding for the Nutube device. The end pieces just require this foam to be placed within the outer surround, where the end panels connect to the aluminium body.

The central pieces that cover the window and the button-shaped pieces for the corner-securing holes are not required for the case.

Cut out a piece of foam 38 × 17mm and place this behind the Nutube. This is held between the two 15mm standoffs at the rear of the Nutube.

Note that the enclosure has a specific top and bottom orientation for both the aluminium extrusion and end panels. The front and rear panels have a slightly different profile at the top and bottom edges. While the top edge is straight, the lower edge has a slightly lower moulding below the two left and right corner holes. That matches the same profile on the aluminium extrusion.

Holes need to be drilled for the volume potentiometer and power switch at the front and the DC socket and RCA sockets at the rear. The required front panel hole locations are shown on the label artwork of Fig.14. This can be downloaded in a PDF from the January 2021 page of the *PE* website.

A *small* portion along the top edge of the RCA terminal housing plastic needs to be shaved or filed off because it is slightly too high to fit in the case otherwise. Less than 1mm needs to be removed.

You can place the labels on the inside of the panels, cutting around the outside perimeter of each label. Or you can cut out the smaller-sized inner perimeter so the labels can be affixed to the outside of the end pieces.

If the panel label is to be inside the end panel, a paper label could be used. For the front panel, the central window in the artwork will need to be cut out with a hobby knife, to expose the Nutube.

The RCA sockets should be secured to the rear panel with the self-tapping screw, and with the rear edge of the PCB touching the inside of the rear panel.

You can then slide the PCB into the case on the second slot up from the bottom. Don't forget to attach the GND solder lug to the top corner screw at the rear adjacent to the RCA sockets. The wire end of the solder lug will need to be oriented diagonally inward, so it does not foul the end cap border.

Additionally, the anodising layer on the aluminium is a good insulator. It will need to be scraped off at the point where the solder lug makes contact with the screw entry point to ensure good contact with the metal.

Finally, the rubber feet provided with the enclosure can now be fixed to the base using their self-adhesive.

Parts list – Nutube Valve Preamplifier

1 double-sided PCB coded 01112191, 98 x 114mm from the *PE PCB Service*
1 set of front and rear panel labels (see text)
1 extruded aluminium enclosure with clear end panels, 115 x 51 x 119mm [Jaycar HB6294]
1 Korg Nutube 6P1 double-triode thermionic valve (V1) [RS Components 144-9016]
1 1A DPDT 5V relay (RLY1) [Altronics S4147]
1 SPDT sub-miniature toggle switch (S1) [Altronics S1421]
1 double stereo horizontal PCB-mount RCA socket assembly (CON1) [Altronics P0211]
1 PCB-mount DC power socket (CON2) [Jaycar PS0520, Altronics P0621A]
1 2-pin 2.54mm-pitch vertical polarised header (CON3) [Jaycar HM3412, Altronics P5492]
1 inline plug to suit CON3 [Jaycar HM3402, Altronics P5472 + P5470A x 2]
1 3-way pin header, 2.54mm pitch with shorting block (JP1)
2 5mm-long ferrite RF suppression beads, 4mm outer diameter (FB1,FB2) [Altronics L5250A, Jaycar LF1250]
1 9V battery
1 9V battery clip with flying leads
1 13-16mm diameter knob to suit VR1
1 8-pin DIL IC socket (optional)
1 100mm cable tie
4 15mm-long M3 tapped spacers
2 M3 x 25mm nylon or polycarbonate panhead machine screws
4 M3 x 6mm panhead machine screws
2 M3 hex nuts
1 No.4 x 8mm self-tapping screw
1 90mm length of medium-duty hookup wire
1 solder lug
4 PC stakes

Semiconductors
3 OPA1662AID dual op amps, SOIC-8 (IC1-IC3) [RS Components 825-8424]
1 LM358 dual op amp, DIP-8 (IC4)
1 TPS70960DBVT 6V regulator, SOT-23-5 (REG1) [RS Components 900-9876]
1 BC337 NPN transistor (Q1)
2 1N4148 small-signal diodes (D1,D2)
1 1N4004 1A diode (D3)
1 1N5819 1A schottky diode (D4)

Capacitors
3 100µF 25V PC electrolytic
2 47µF 16V PC electrolytic
10 10µF 25V PC electrolytic
1 2.2µF X7R SMD ceramic, 2012/0805 package [RS Components 6911170]
2 470nF MKT polyester
6 100nF MKT polyester
2 100pF ceramic

Resistors (all 0.25W, 1% metal film)
5 1MΩ
2 330kΩ
1 180kΩ
10 100kΩ
2 33kΩ
4 10kΩ
1 6.8kΩ
3 5.1kΩ
2 2.2kΩ
2 270Ω
4 150Ω
2 100Ω
1 33Ω
1 dual-gang logarithmic 50kΩ 9mm PCB-mount potentiometer (VR1) [Jaycar RP8760]
2 10kΩ horizontal 5mm trimpots (VR2,VR3)
1 10kΩ top-adjust multiturn trim pot 3296W style (VR4)

Free audio signal generator and analyser software

If you want an audio signal generator that runs on a computer, you can use the free Audacity software (**www.audacityteam.org**). This is available for Windows, macOS, GNU/Linux and other operating systems. Download and install the version that suits the operating system on your computer. Once installed and running, select Generate -> Tone and then set the waveform to sine, frequency to 1kHz and volume to maximum (ie, set the level value to one). You can also set the duration over which the tone is generated. Press the play button for the audio to start.

Another good, easy-to-use option is WaveGene (**http://bit.ly/pe-jan21-wg1**).

For spectrum analysis, you could use WaveGene in combination with WaveSpectra (**http://bit.ly/pe-jan21-wg2**). See the setup instructions at: **http://bit.ly/pe-jan21-wg3**

We used Visual Analyser, available from **http://bit.ly/pe-jan21-va**, mainly because this allows the actual measured waveform to be seen as a 'scope' view, along with the output spectrum.

Once you have installed the signal generator and spectrum analyser software, it's a good idea to use it to analyse the performance of your computer sound interface. That can be done with a cable with 3.5mm stereo jack plugs at each end, with one end plugged into the sound input and one into the sound output.

To do this with Visual Analyser, on the main screen, select 'floating windows mode' and then the Scope, Spectrum and Wave need to be opened from the top row of selections.

Select a 1kHz sinewave for the wave generator, select interlock (that causes both A and B channels to change together) for the output levels and bring up the output level on the waveform generator. Then press the on/off button below the output level slider.

The on/off selection at the top left of the main screen also needs to be selected so that the analyser measures the signal. Both will show 'off' when the signal is generated and measured. You can choose to view the A channel (left) or B channel (right), or both, in the main settings channel selection.

We chose to use a 16,384 sample FFT window and a sampling rate of 44.1kHz in the main menu. Output gain (adjustment along the top row at right) was set just below maximum, yielding the lowest distortion figure of 0.0626%.

In our case, noise is mostly more than 80dB *below* the fundamental (see Fig.11). That indicates that this is not a particularly good sound card, but good enough to evaluate the distortion from the *Nutube Preamplifier.*

Now the *Nutube Preamplifier* can be connected between the computer sound input and output. Adjust signal levels using the volume control and/or the signal generator level so that the waveform is not clipped (ie, so the top of the sinewave is not plateauing) and shows a clean sinewave.

In the main menu, you can select the left channel (A) and adjust trimpot VR2 for the lowest distortion reading, with minimal harmonics – see Fig.12. This shows the waveform as a clean sinewave, with the analyser showing the main 1kHz fundamental at 0dB level and the second harmonic (2kHz) at around –70dB. The third, fourth and sixth harmonics are at a similar level.

Once you've finished tweaking VR2, select the right channel (B) and adjust VR3 for the lowest distortion reading.

VR4 can then be adjusted while viewing in the A channel of the analyser, so that fundamental level is the same as that in the B channel.

Fig.13 shows the waveform and spectrum when the grid bias (with VR2) is adjusted *incorrectly.* The top half of the sine waveform is very rounded, and the second harmonic is only 10dB below the fundamental. The distortion reading is around 30%.

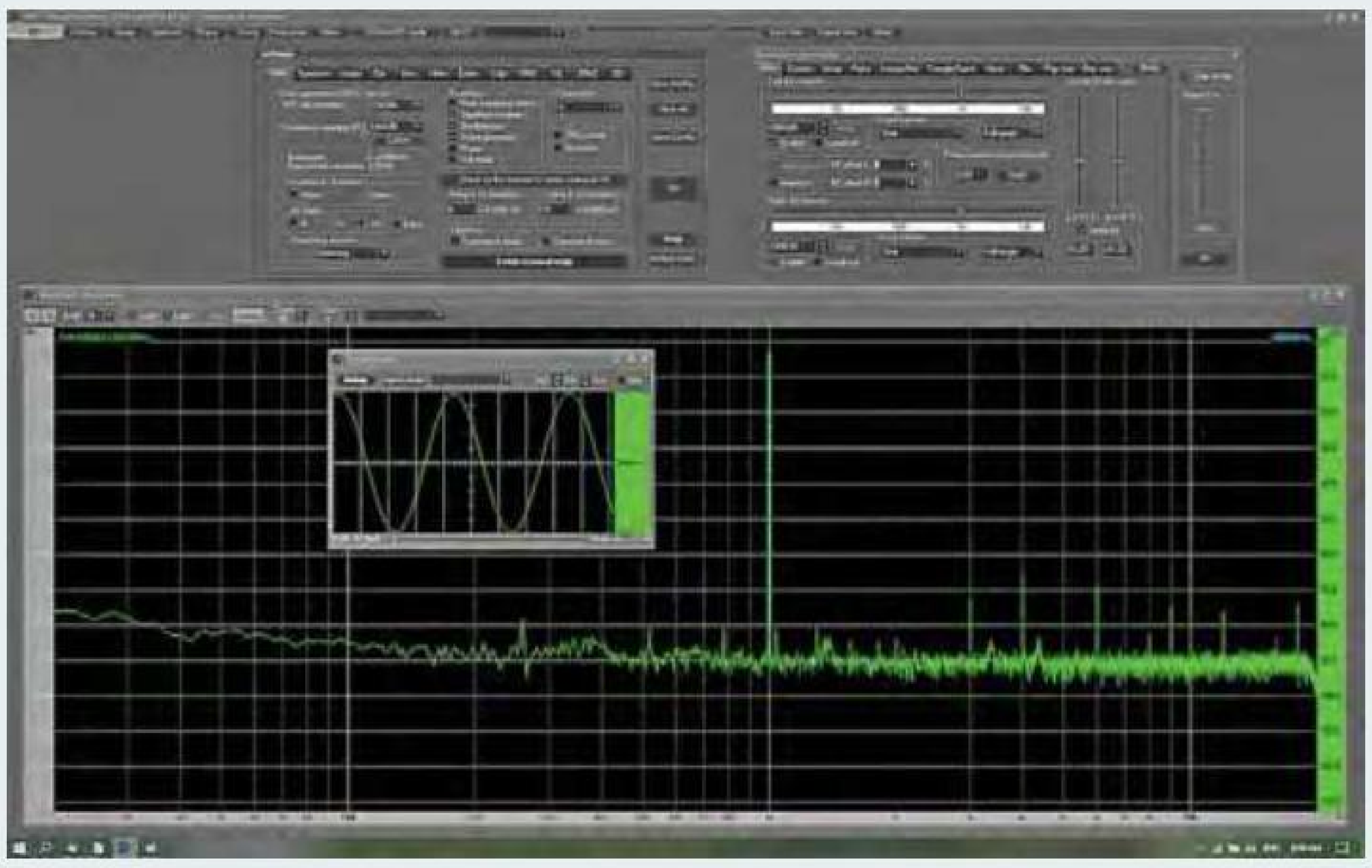


Fig.11: a screen grab of the free Visual Analyser PC software performing a 'loopback' test, with the sound card output fed directly into its input. This lets you analyse the distortion inherent in the system. In this case, the reading is 0.0626% THD+N at 1kHz. You therefore won't get a reading lower than that when measuring the performance of external devices like the *Nutube Preamplifier*.

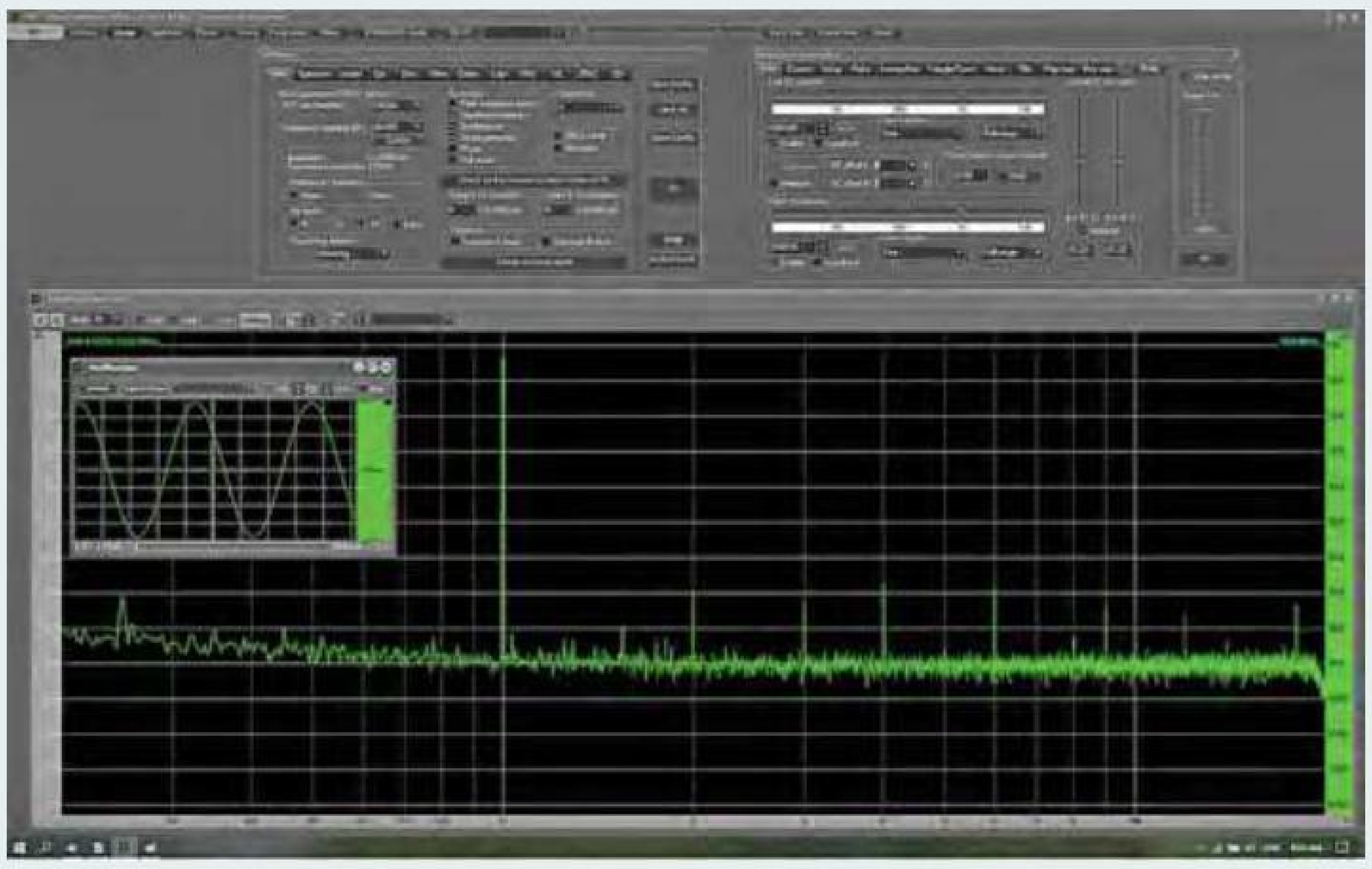


Fig.12: now we have connected the *Nutube Preamplifier* 'in the loop' between the sound card output and input, using two stereo jack plug to red/white RCA plug cables. The output levels have been set to 41% full-scale, which corresponds to around 250mV RMS, The distortion reading has only risen slightly, to 0.07%, because the *Nutube Preamplifier* and sound card distortion figures are similar.

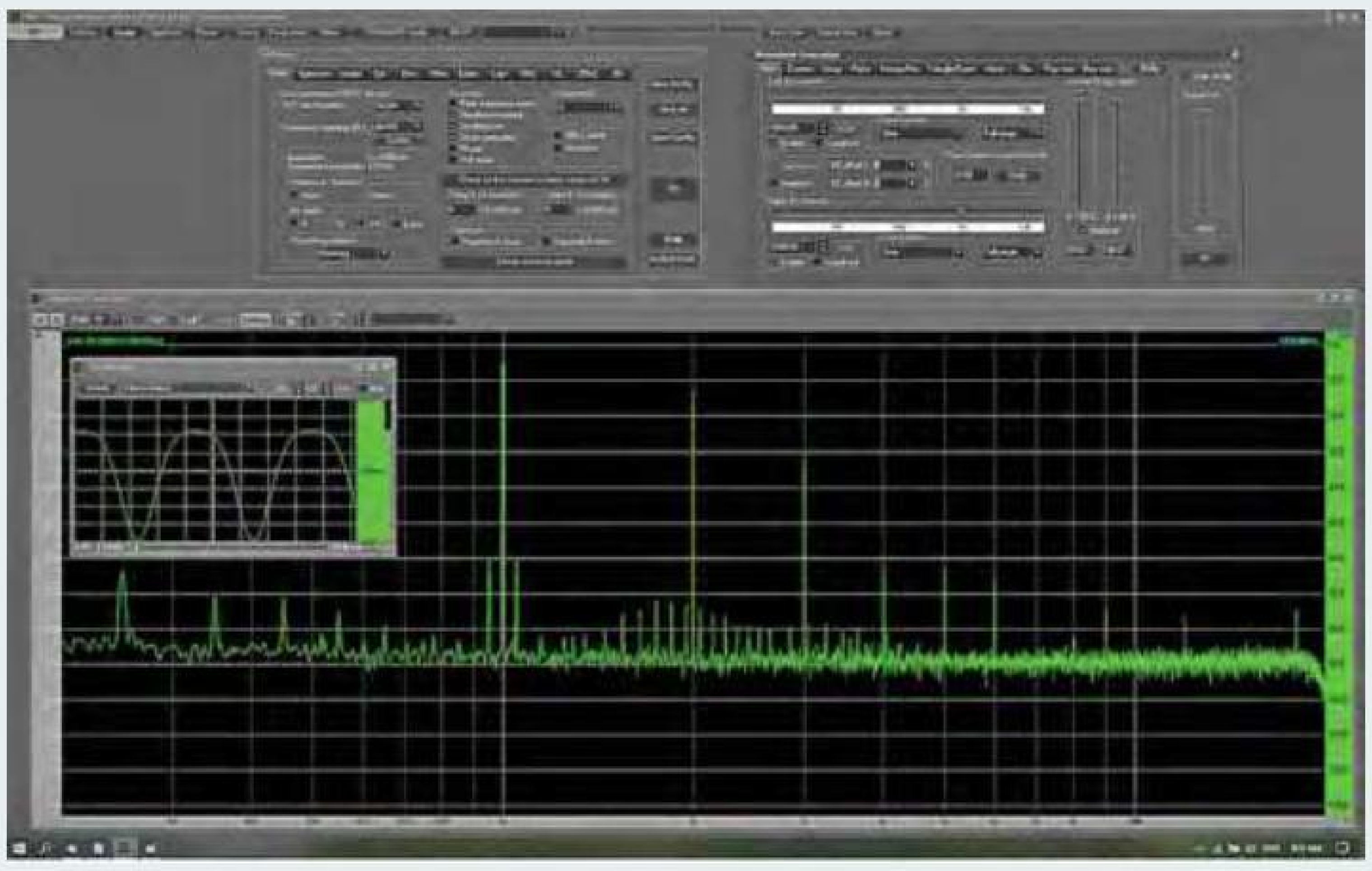
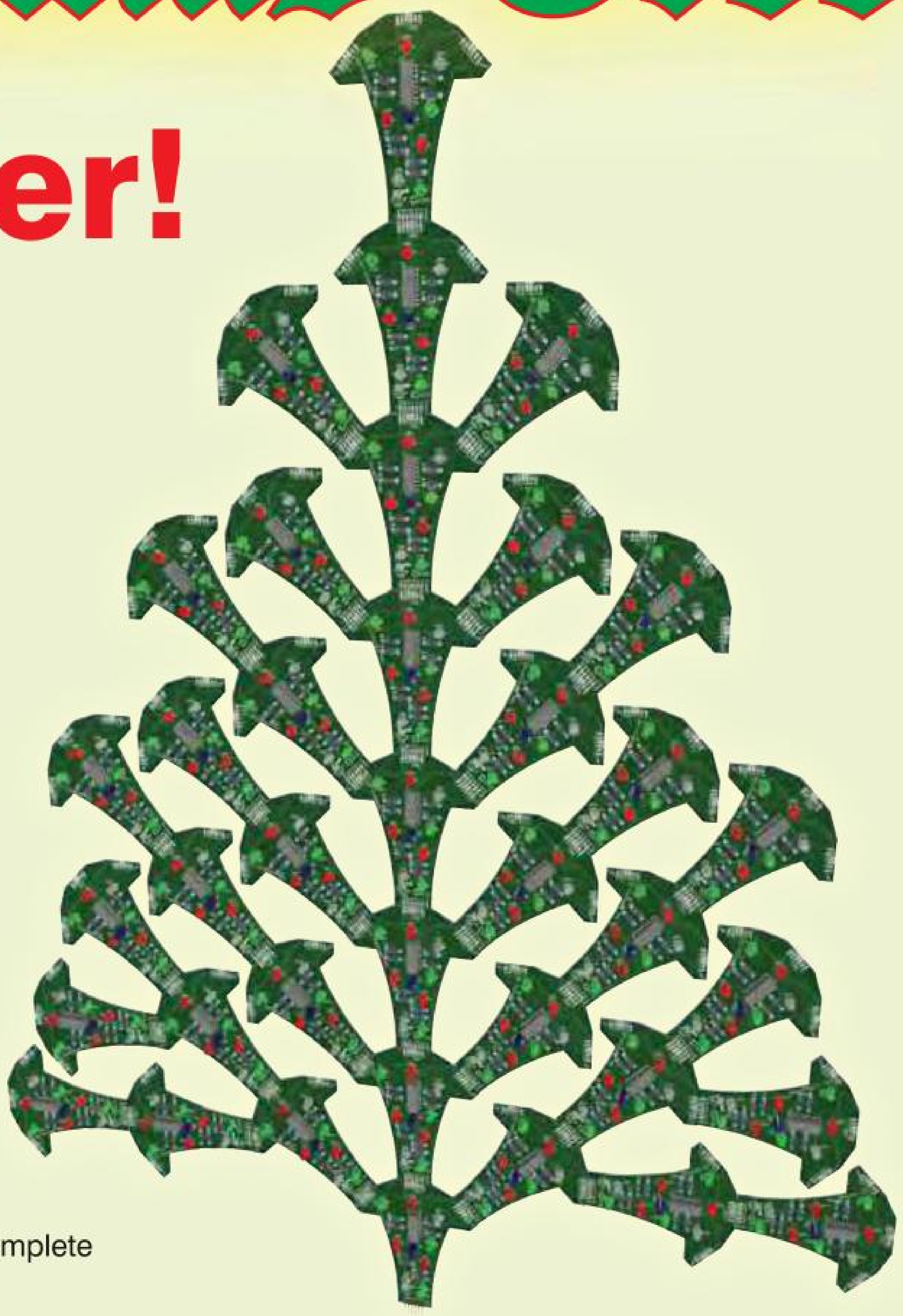


Fig.13: here is the same test as Fig.12, but the triode grid bias voltage adjustment is completely wrong. You can see the heavily distorted sinewave in the 'Oscilloscope' window, with many harmonics in the spectrum analysis. The THD reading is 30%. This is about as bad as it gets; more realistically, a slightly misadjusted grid bias voltage can lead to distortion levels in the 0.1-1% range.

Reproduced by arrangement with
SILICON CHIP magazine 2020.
www.siliconchip.com.au

LED Christmas Tree



PCB special offer!

Here's a little Christmas bargain to help you build your very own stackable LED Christmas Tree decoration.*

Buy a single LED Christmas Tree PCB for £6.95

4 PCBs costs just £14.95

12 PCBs costs just £24.95

20 PCBs costs just £34.95

Visit our shop at www.electronpublishing.com and place your order!

Want even more? Just email us for a quote: shop@electronpublishing.com

*Please note this is not a kit of parts – you need to supply your own components to complete the project. Full construction details in the November 2020 issue of *PE*.

BACK ISSUES

OCTOBER 2020

PROJECTS • High-Power 45V/8A Variable Linear Supply – Part 1 • Precision 'Audio' Signal Amplifier • Arduino-based Digital Audio Millivoltmeter • Pedal Power Station! – Part 1

FEATURES • The Fox Report • Techno Talk • Net Work • Circuit Surgery • Audio Out • Practically Speaking • Make it with Micromite • Max's Cool Beans

NOVEMBER 2020

PROJECTS • The Christmas Tree that grows! • USB Digital and SPI Interface Board • High-Power 45V/8A Variable Linear Supply – Part 2 • Five-way LCD Panel Meter / USB Display • Pedal Power Station! – Part 2

FEATURES • The Fox Report • Techno Talk • Net Work • Circuit Surgery • Audio Out • Practically Speaking • Make it with Micromite • Max's Cool Beans

DECEMBER 2020

PROJECTS • Clever Controller for a Dumb Battery Charger • LFSR Random Number Generator • High-Power 45V/8A Variable Linear Supply – Part 3 • Building a Hi-Fi amp on the cheap

FEATURES • The Fox Report • Techno Talk • Net Work • Circuit Surgery • Audio Out • Make it with Micromite • Max's Cool Beans

BACK ISSUES – ONLY £5.80

£5.80 per issue for UK incl p&p ■ £6.75 Europe Air Mail ■ £7.50 ROW Air Mail

We can supply back issues of PE/EPE by post.

We stock magazines back to 2006, except for the following:

2006 Jan, Feb, Mar, Apr, May, Jul
2007 Jun, Jul, Aug
2008 Aug, Nov, Dec
2009 Jan, Mar, Apr
2010 May, Jun, Jul, Aug, Oct, Nov
2011 Jan
2014 Jan
2018 Jan, Nov, Dec
2019 Jan, Feb, Apr, May, Jun

Issues from Jan 1999 are available on CD-ROM / DVD-ROM

If we do not have a a paper version of a particular issue, then a PDF can be supplied for the same price. Your email address must be included on your order.

Please make sure all components are still available before commencing any project from a back-dated issue.

ORDER FORM – BACK ISSUES

☐ Back issues required (month / year) ....
Name ....
Address...
....Tel:...
Email ....
☐ I enclose cheque/PO to the value of .... £............
☐ Please charge my Visa/Mastercard .... £...............
Card No ....
Valid From .... Card Expiry Date ....
Card Security Code .... (Last three digits on or under the signature strip)

SEND TO: **Practical Electronics, Electron Publishing Limited**
113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU

Tel: 01202 880299 **Email**: stewart.kearn@wimborne.co.uk **On-line Shop**: www.electronpublishing.com

Payments must be in £ sterling – cheque must be drawn on a UK bank and made payable to 'Practical Electronics'. All items normally posted within seven days of receipt of order. Copy this form if you do not wish to cut your issue.

Complete Arduino DCC Controller

Digital Command Control (DCC) is a great way to control multiple trains on a model railway layout. Unfortunately, commercial DCC systems can be quite expensive. Here we present an Arduino-compatible controller shield that can form the basis of a DCC system. It can also be used as a DCC booster or even as a high-current DC motor driver.

by Tim Blythman

You can put together this DCC *Controller*, which incorporates a *Base Station* and optionally also a programmer, for a fraction of the price of a commercial unit. Combine it with a PC, and you have a potent and flexible model railway control system.

It's based on the Arduino platform, and it's easy to build. You can also add boosters to the system easily, just by building a few more shield boards.

DCC is still the 'state-of-the-art' in terms of off-the-shelf model railway systems, so if you have a model railway layout but don't have a DCC system (or have a DCC system that's inadequate for your needs), now is the time to upgrade!

We published an *Arduino-based DCC Programmer for Decoders* in our October 2019 issue. Since then, we've had numerous requests for a DCC base station or booster.

Therefore, we have created this DCC power shield, which is the final piece of the puzzle.

Adding this (and an appropriate power supply) to the *Programmer*, in conjuction with DCC-capable locos, results in a complete DCC system.

As this is an Arduino-based project, the following description assumes that you are familiar with the Arduino IDE (Integrated Development Environment).

To download the free IDE software, go to: **www.arduino.cc/en/software**

We are using version 1.8.5 of the IDE for this project, and suggest that if you have an older version installed, that you upgrade it now.

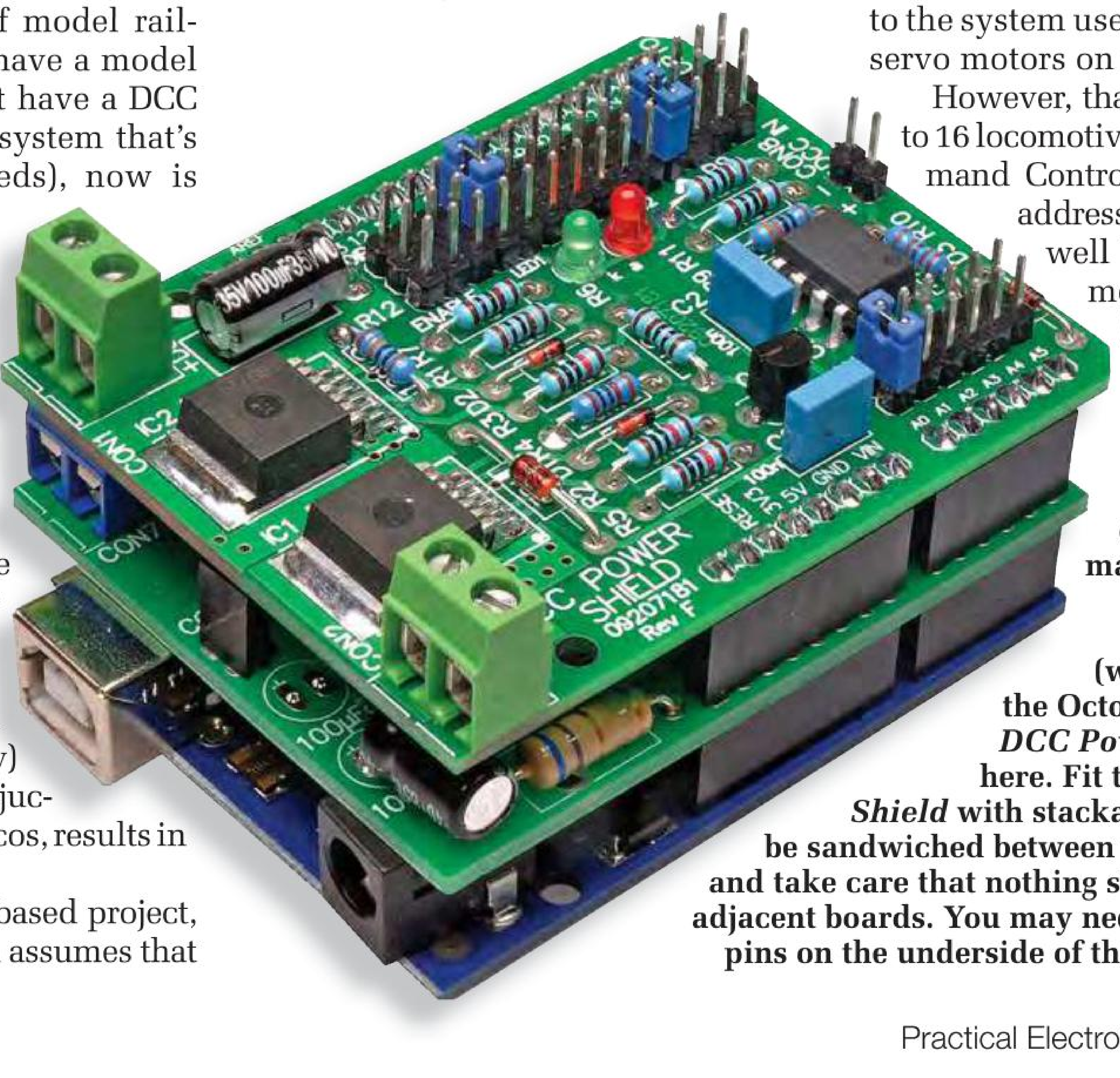
What is DCC?

We went into a bit of detail on DCC in the *DCC Programmer* article, so we'll only cover the basics here. If you want to learn more, read the aforementioned article from October 2019.

DCC is designed to allow multiple model trains to be controlled on a single track, with the same set of tracks carrying power for the trains and also digital control commands.

Older command controls systems exist; for example, the Protopower 16, which was based on another system called CTC16. This worked similarly to the system used to control multiple servo motors on model aircraft.

However, that system was limited to 16 locomotives, while Digital Command Control has around 10,000 addresses available; probably well beyond the scope of most model railroads (and many full-scale railroads too!).



A complete DCC control system can be made by adding an Uno board and the *DCC Programmer Shield* (which we described in the October 2019 issue) to the *DCC Power Shield*, as shown here. Fit the *DCC Programmer Shield* with stackable headers, so it can be sandwiched between the other two boards, and take care that nothing shorts out between the adjacent boards. You may need to trim some of the pins on the underside of the *DCC Power Shield*.

Two locos, one track – but both are under individual control of the DCC system. As you can *just* see, the loco in front even has its headlight on – also switched on or off at will via DCC. Want more than two trains? DCC has up to 10,000 addresses available!

The most basic method of model train control is for a single throttle to apply a variable DC voltage to the track, which drives the train's motor directly. Instead, a DCC base station delivers a high-frequency square wave to the track. The base station encodes binary control data into this signal by varying the width of each pulse (see Fig.1).

A digital decoder on each vehicle receives commands and also rectifies the AC track voltage to produce DC. The decoder then uses this to drive the motor and can also control lights, sound effects (like a horn or engine) or even a smoke generator.

There are also accessory decoders which can be used to control things such as points and signals using the same DCC signals.

The DCC standard is produced by the National Model Railroad Association (based in the US; see **http://bit.ly/pe-jan21-dcca**). These standards are available for download, which means that anyone can use them. As a result, many different manufacturers are making DCC-compatible equipment.

Our *Base Station* will work with many commercially-available decoders. There is a vast array of manufacturers of DCC equipment, so we can only test a small subset. All of those we have tested have worked well, as should be expected from a proper application of the standard.

Terminology

A *Base Station* – in DCC terminology – is the brains of the system. Typically, it receives commands from attached throttles controlled by people, or perhaps a computer. These commands then dictate what DCC data needs to be sent over the tracks to the trains to control them.

The *Base Station* generates a continuous stream of DCC data packets to control and update all trains, signals and points as needed.

A *Booster* is a simple device which takes a low-level DCC signal and produces a DCC signal of sufficient power to drive a set of tracks. Many smaller DCC systems consist of a single unit which combines a *Base Station* with a *Booster*, while larger systems might have separate units, including multiple *Boosters*.

Our *DCC Power Shield* works as a *Booster*. An attached and properly programmed Arduino board can be used as the *Base Station* smarts, thus creating a basic DCC system in a single unit. Extra *DCC Power Shields* can be deployed as separate *Boosters*, with an Arduino attached to monitor each and check for faults.

When programmed with the DCC++ software, the Arduino board and *DCC Power Shield* can be combined with our earlier *DCC Programming Shield* to create a compact, economical and fully-featured DCC system.

Power source

A DC power source is needed to run the *DCC Power Shield*. The DCC standards suggest that *Boosters* should produce 12V-22V peak, so your chosen power source needs a regulated DC output in this range.

For modest current requirements (up to around 5A), a laptop power supply is a good choice. Many of these have a nominal 19V DC output at several amps. Any fully DCC-compatible trains and decoders should handle this fine, but it's worth checking any that you aren't sure about.

Decoders are supposed to work down to around 7V. Given that the track, wiring and locomotives are bound to drop some voltage, a 12V 'power brick' type supply works well enough for driving trains. However, we found that this sometimes wasn't enough to allow decoder programming to occur.

If you need more current than a laptop power supply can provide, you will need to find a dedicated power supply in the 12-22V range. Many suitable high-power 'open frame' switchmode supplies are available from various suppliers.

One thing to note is that while some Arduino boards (including genuine boards) can tolerate up to 20V on their VIN inputs, some clones use lower-rated voltage regulators which can only handle 15V.

We have provided an option for a zener diode to help manage this variation; read on for more information on how the circuit works.

DCC Power Shield circuit

The circuit of the *DCC Power Shield* is shown in Fig.2. Its key function is to turn a steady DC voltage into a DCC-modulated square wave. For this, we need a full H-bridge driver. To keep it simple, we have used a pair of BTN8962 half-bridge driver ICs (IC1 and IC2).

Features and specifications

- *Based on the Arduino Uno*
- *Provides a DCC output of 12-22V peak at up to 10A, or more with some changes*
- *Can operate as a Base Station or Booster*
- *Compatible with DCC++ and JMRI (DecoderPro/PanelPro) software*
- *Opto-isolated input for use as DCC slave*
- *Works with our DCC Programmer shield from the October 2019 issue*
- *Can also be used as a brushed-motor driver*
- *All Arduino pin assignments configurable via jumpers*

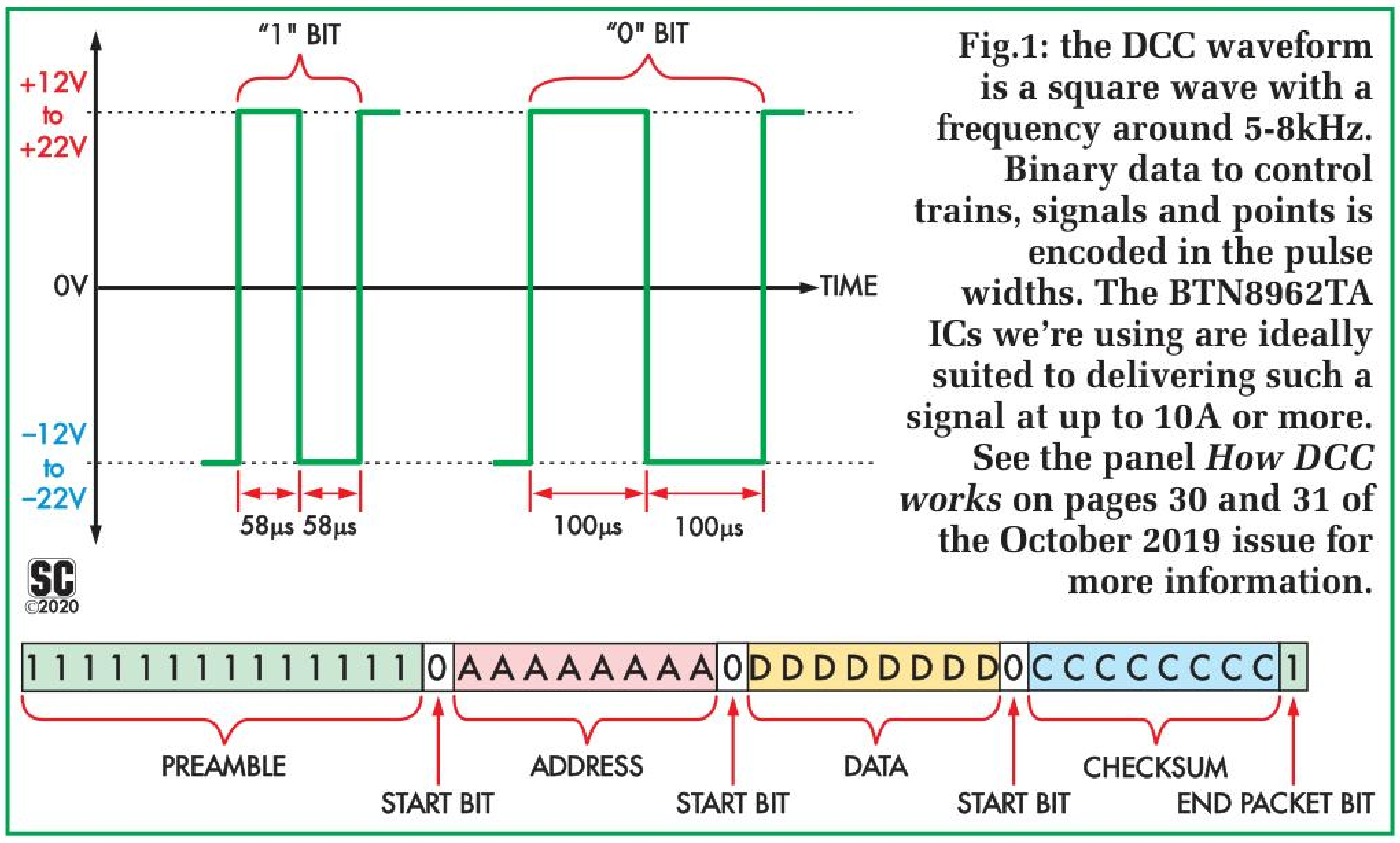


Fig.1: the DCC waveform is a square wave with a frequency around 5-8kHz. Binary data to control trains, signals and points is encoded in the pulse widths. The BTN8962TA ICs we're using are ideally suited to delivering such a signal at up to 10A or more. See the panel *How DCC works* on pages 30 and 31 of the October 2019 issue for more information.

The BTN8962 comes in a TO-263-7 package, which is a surface-mounting part, although quite a large one. It is not difficult to solder. There are two of these, one driving each side of the track. They are supplied with out-of-phase input signals to produce the required alternating output drive.

Their supply pins (pins 1 and 7) are connected directly across the incoming DC supply from CON1, labelled VIN.

A 100µF electrolytic capacitor bypasses this supply. 100µF may seem like a low value, the current drawn by IC1 and IC2 is quite steady as when one output goes high, at the same time, the other goes low. The outputs of IC1 and IC2 connect to screw terminal CON2, and then onto the tracks.

The state of the IN pins (pin 2) determines whether the output pins (4 and 8) are driven high or low. The SR input pin controls the output slew rate. We've tied this to ground to give the fastest possible slew rate. The 'INH' pins (pin 3) need to be brought high to enable the outputs. These are connected together and have a 100kΩ pull-down resistor so that the outputs default to off.

The enable signal connects back to an Arduino pin via a 10kΩ resistor and jumper JP1, allowing the Arduino to enable or disable the outputs as required. JP1 lets any Arduino digital pin connect to the enable signal, to suit the software used.

The IS pins (pin 6) on IC1 and IC2 are outputs that source a current proportional to the current being drawn from the output of each IC (plus a small offset current, which is compensated for in software). These currents are combined in a 'diode-OR' circuit formed by diodes D1 and D2 and then fed to a 1kΩ resistor to convert the combined current into a voltage.

This then passes to an RC low-pass filter (20kΩ/100nF) for smoothing. The 2ms time constant means that peaks in the current due to the rapidly changing DCC signal are ignored, but faults can still be detected quickly. The resulting smoothed voltage is fed to one of the Arduino analogue input pins via jumper JP2, to allow the Arduino to monitor the track current.

JP2 allows any of the Arduino analogue inputs to be used to monitor track current, again allowing us to choose whichever pin suits the Arduino software.

The IS pins will also source current if IC1 or IC2 detects an internal fault condition; as far as the software is concerned, this is equivalent to a very high current being drawn from the output and is treated the same way.

Bridge driving signals

The input signal to pin 2 of IC2 comes from another one of the Arduino digital outputs via a 10kΩ series resistor. Once again, any Arduino digital pin can be used, and this too is selected by a jumper shunt on JP1.

A simple inverter circuit produces the out-of-phase signal to drive the IN pin of IC1. The signal that goes to pin 2 of IC2 is also fed to the base of NPN transistor Q1 via a 1kΩ resistor. Q1's collector is pulled up by a 10kΩ resistor to the ENABLE line. So as long as ENABLE is high, meaning the outputs of IC1 and IC2 are active, input pin 2 of IC1 is inverted compared to input pin 2 of IC2.

Opto-isolated input

To allow a separate *Base Station* to be used, an optoisolated input is provided at CON3. This can accept a logic-level DCC signal, or even a 'track voltage' (12-22V) signal from another DCC system.

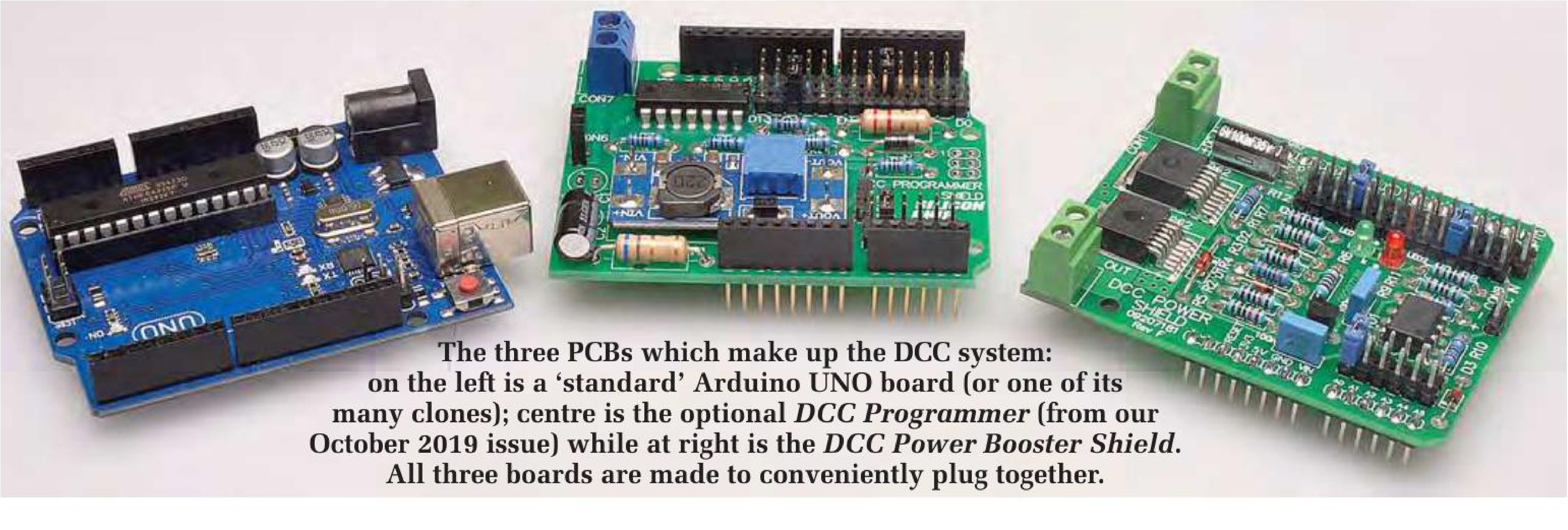
The signal at CON3 passes through a 2.2kΩ series resistor and into the LED of OPTO1. 1N4148 diode D3 is connected in reverse across this LED, to protect it from high reverse voltages.

If a logic-level DCC signal is applied to CON3, then the polarity markings need to be observed, as current will only flow through OPTO1 when the voltage at pin 2 is high. A bipolar DCC signal can be connected either way around.

OPTO1 is a 6N137 high-speed opto-isolator which has a nominal forward current of 10mA. Thus the 2.2kΩ resistor is suitable for voltages up to around 22V, ie, the maximum expected from a DCC system.

The output of OPTO1 is supplied with 5V from the Arduino board, bypassed by a 100nF capacitor. A 330Ω pull-up resistor sets the logic-high level.

The output from OPTO1's pin 6 is fed via a 1kΩ protection resistor to jumper JP3. This allows the DCC signal to be fed directly to the input of bridge drivers IC1 and IC2.



The three PCBs which make up the DCC system: on the left is a 'standard' Arduino UNO board (or one of its many clones); centre is the optional *DCC Programmer* (from our October 2019 issue) while at right is the *DCC Power Booster Shield*. All three boards are made to conveniently plug together.

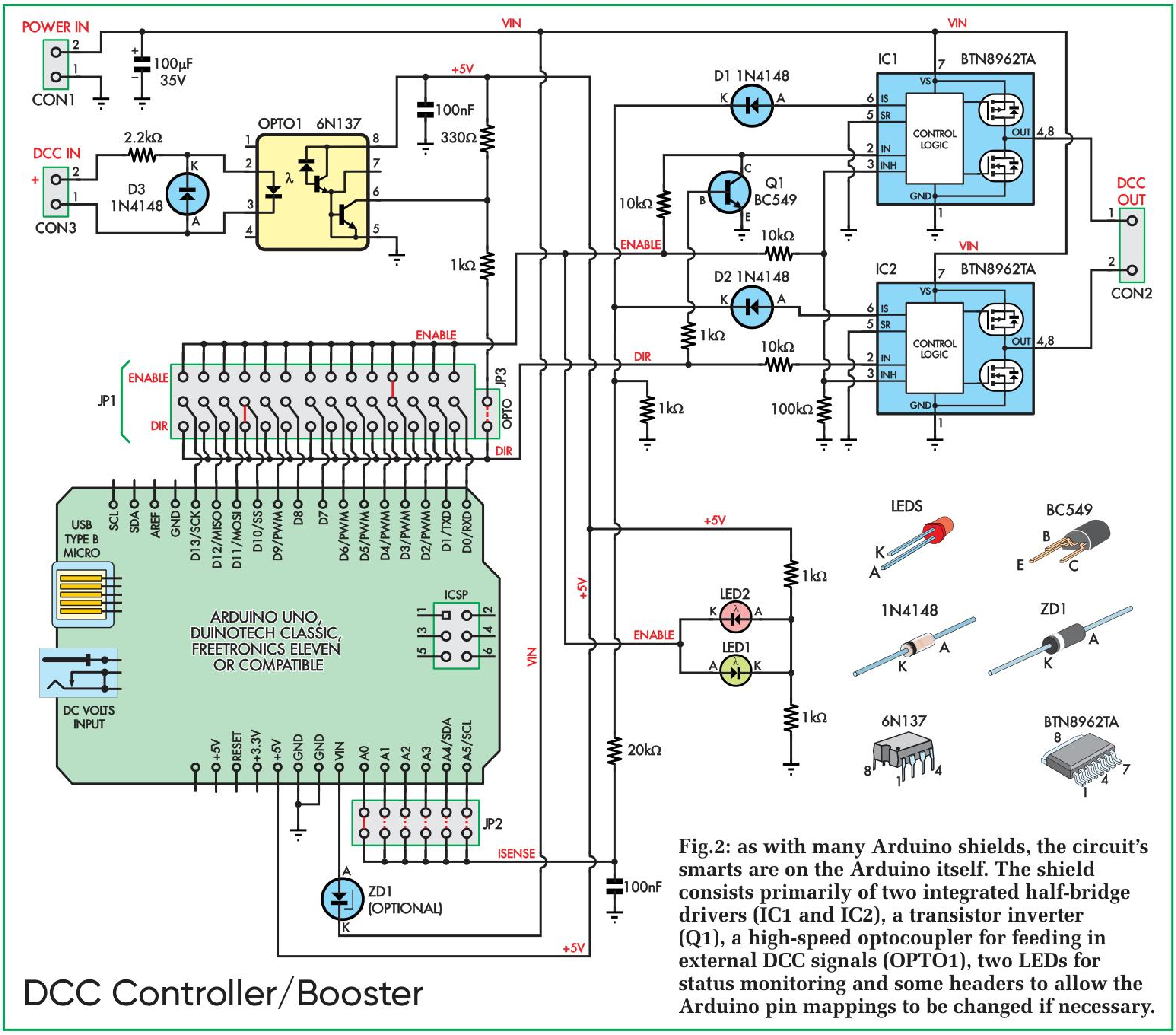

POWER IN
CON1
100µF 35V
VIN
+5V
100nF
330Ω
OPTO1 6N137
DCC IN
CON3
2.2kΩ
D3 1N4148
1kΩ
ENABLE
DIR
JP1
JP3
OPTO
IC1 BTN8962TA
IC2 BTN8962TA
VS
IS
SR
IN
INH
GND
OUT
CONTROL LOGIC
D1 1N4148
D2 1N4148
Q1 BC549
10kΩ
100kΩ
DCC OUT
CON2
USB TYPE B MICRO
DC VOLTS INPUT
SCL
SDA
AREF
GND
D13/SCK
D12/MISO
D11/MOSI
D10/SS
D9/PWM
D8
D7
D6/PWM
D5/PWM
D4/PWM
D3/PWM
D2/PWM
D1/TXD
D0/RXD
ARDUINO UNO, DUINOTECH CLASSIC, FREETRONICS ELEVEN OR COMPATIBLE
ICSP
+5V
RESET
+3.3V
A0
A1
A2
A3
A4/SDA
A5/SCL
JP2
ISENSE
ZD1 (OPTIONAL)
20kΩ
LED2
LED1
LEDS
BC549
1N4148
ZD1
6N137
BTN8962TA
DCC Controller/Booster

Fig.2: as with many Arduino shields, the circuit's smarts are on the Arduino itself. The shield consists primarily of two integrated half-bridge drivers (IC1 and IC2), a transistor inverter (Q1), a high-speed optocoupler for feeding in external DCC signals (OPTO1), two LEDs for status monitoring and some headers to allow the Arduino pin mappings to be changed if necessary.

In this case, a jumper on JP1 can be used to feed the same signal to one of the Arduino's digital pins, which would then be configured as an input.

Due to the open-collector output of OPTO1, this signal is inverted compared to that applied to CON3.

But this can be solved simply by reversing the connections from CON2 to the tracks.

This reversibility of the DCC signal is a necessary feature, as a locomotive may be placed on the track either way and must be able to work with an inverted signal.

The only time this matters is when different boosters feed two adjoining tracks. In that case, you will need to make sure that the signals are in-phase.

Other features

Status LEDs LED1 and LED2 are connected to the ENABLE signal with 1kΩ current-limiting resistors to GND and 5V respectively.

So if ENABLE is high, the green LED1 lights up, and if it's low, then the red LED2 lights up instead. If ENABLE is high-impedance, such as when the Arduino is in reset, neither LED lights. A single bi-colour LED could be fitted either for LED1 or LED2 to achieve the same effect.

If fitted, ZD1 feeds DC from CON1 to the VIN input of the Arduino board. Its value is chosen to limit the Arduino input voltage to a safe level at the maximum expected voltage from CON1.

For example, for 22V into CON1, ZD1 can be an 8.2V type, so 13.8V is fed to the Arduino VIN pin. A 1W, 8.2V zener diode can pass up to 120mA, which should be enough to power the Arduino and any connected shields.

We've left enough space to fit a 5W zener diode if you need more current than that, although if you're going to be applying less than 22V to CON1, you could also use a lower-voltage zener, which could then pass more current before reaching its 1W limit.

For situations where the voltage on CON1 is suitable for direct connection to VIN (typically under 15V for clones or 20V for genuine Arduino boards), then a wire link can replace ZD1.

However, it would still be a good idea to fit a low-voltage zener (eg, 3.3V) as this will reduce the dissipation in the Arduino's regulator. Just make sure that the voltage fed to the Arduino's VIN pin will not drop below 7V.

If you aren't sure whether your Arduino can handle more than 15V, check the onboard regulator. It's usually in an SOT-223 three-pin SMD package with a hefty tab.

Genuine Arduino Uno boards usually have an NCP1117 regulator, rated to handle up to 20V. Clones often have an AMS1117 instead, which is only rated to 15V.

If ZD1 is left off, the supplies are separate (although their grounds will be connected). This allows the Arduino to be powered via its USB connector, eg, from a controlling computer.

DCC Programming

Many *DCC Base Stations* have a separate output for programming decoders.

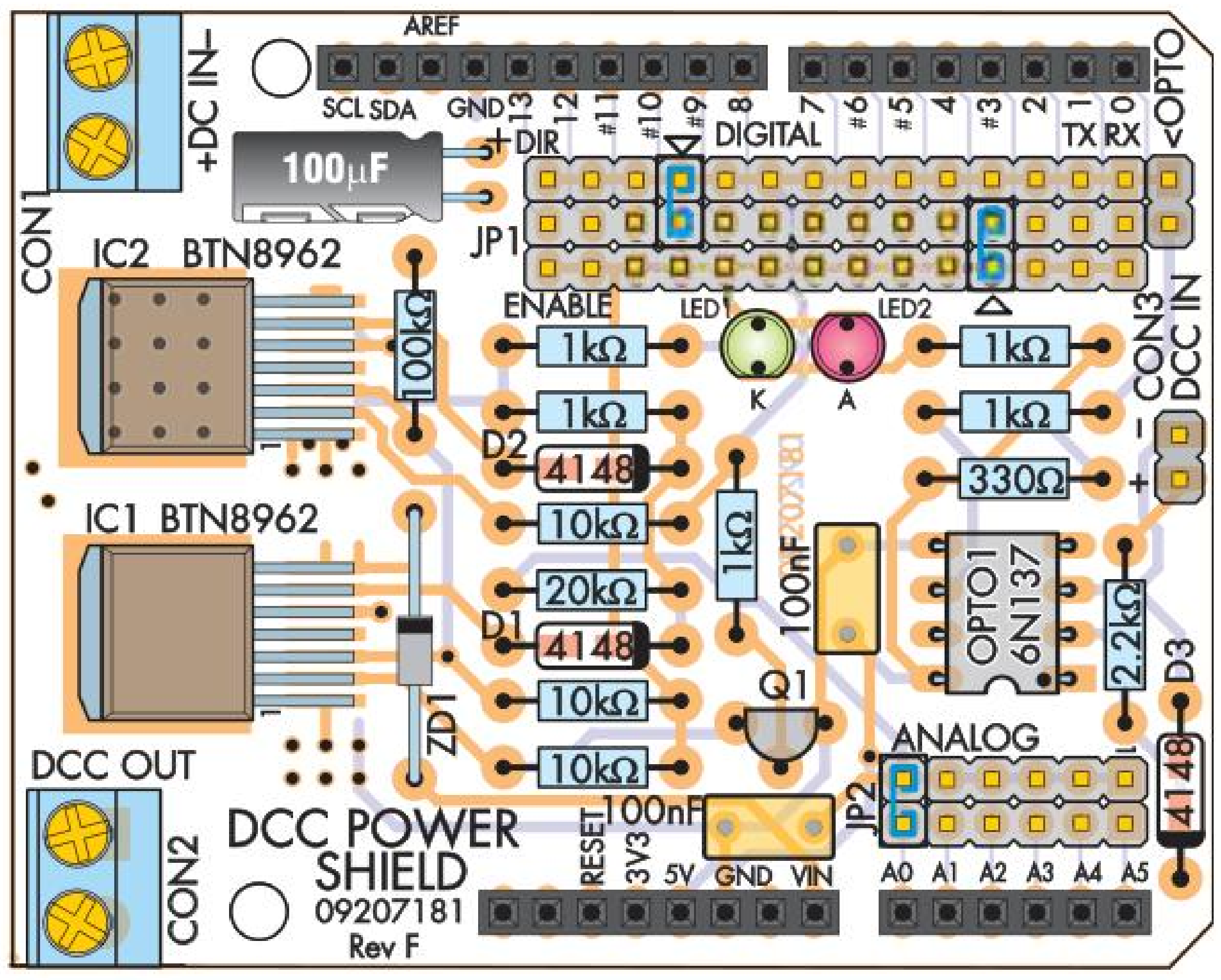
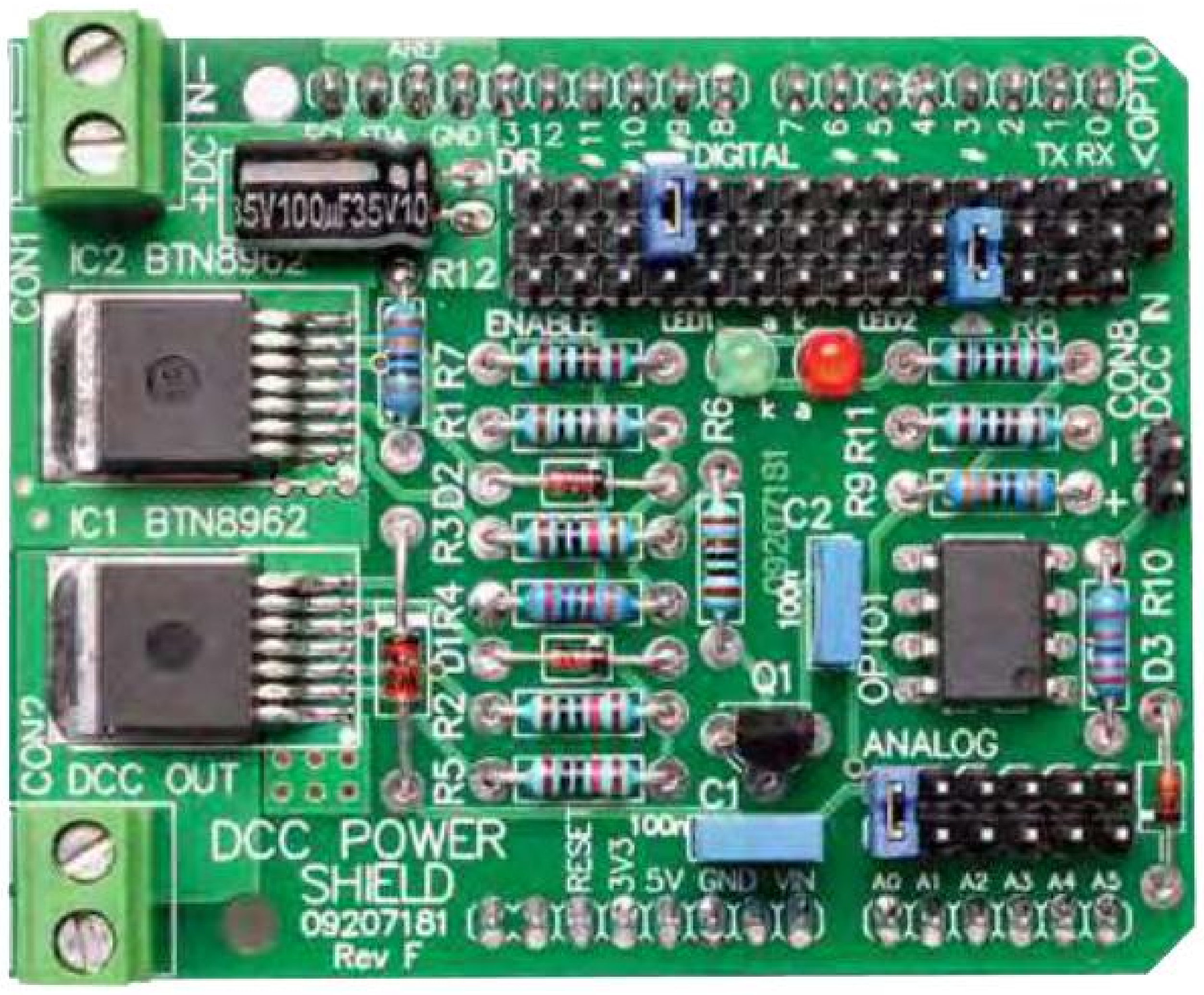


Fig.3: the seven-pin half-bridge driver ICs are mounted on the left, near the power input (CON1) and track (CON2) terminals. The jumper positions shown here are those required to use both the open-source DCC++ software and our example sketches. The jumpers are mostly handy if you want to use this shield as a DC motor driver, so that you can connect the required functions to PWM pins.



In other words, programming is not done via the main high-current output driver, which is usually kept connected to the layout.

For this reason, you may wish to have the *DCC Power Shield* and October 2019 *DCC Programmer* shield plugged into the same Arduino. The DCC++ software is designed to handle this.

However, this does complicate the power supply arrangements a bit.

First, the *DCC Programmer Shield* has a maximum supply voltage of 15V, so regardless of the type of Arduino board you are using, you will need to ensure that the VIN pin is no higher than 15V.

Also, in this system, it would be best to build the *DCC Programmer Shield* without the MT3608 boost module, and fit the jumper shunt on CON8 between pins 1 and 2, so that the VIN supply is used for programming power.

The *DCC Programmer Shield* can draw up to 200mA from VIN, so the dissipation of ZD1 will increase substantially. You will need to choose its value carefully, or use a 5W zener.

Another option, if the system will always be connected to a computer, is to build the *DCC Programmer Shield* with the MT3608 boost module and fit it below the *DCC Power Shield*, then leave out ZD1 from the *Power Shield*.

The *DCC Programmer Shield* will then be powered from the computer's 5V USB supply, while the *DCC Power Shield* is still powered via CON1.

Construction

The *DCC Power Shield* is built on a double-sided PCB in a typical Arduino shield shape, coded 09207181, measuring 68.5 × 55mm and available from the *PE PCB Service*. Use the overlay diagram, Fig.3, as a construction guide .

Start by fitting IC1 and IC2. Although these are surface-mounting components, they are quite large. Because of this, and the fact that they sit on large copper pours, it will require quite a bit of heat to make good solder joints.

Flux paste and solder braid (wick) will come in handy, as will tweezers. Apply some flux paste to the pads first, to make soldering easier.

Working on one at a time, start by tacking one of the end pins in place to locate the device.

Once you are happy that each is centrally located within the footprint, load some solder on the tip of your iron and apply it to each of the smaller pads. Ensure that the resulting solder fillets are solid.

Use the solder braid to remove any solder bridges. The two end pins, numbers 1 and 7, are ground and power respectively. It's a good idea to add a bit of extra solder to these pins to help with current and heat handling.

Finally, solder the large tab of each device. Hold the iron tip at the point where the tab meets the pad on the PCB. Heat the pad until it melts solder applied to it. Feed in solder until a rounded, but not bulging fillet is formed and allow it to cool.

Next, fit the 12 resistors. The PCB silkscreen is marked with the values, and you should check these match with a multimeter as they are fitted, to ensure they are the correct value. Solder close to the PCB, then trim the leads close to the underside.

Then install the three small 1N4148 diodes (D1-D3) where shown (Fig.3) ensuring that they are correctly oriented.

If fitting ZD1, do that now. Make sure that its cathode band faces towards the top of the PCB. Then mount the rectangular MKT capacitors, which are not polarised.

Now install NPN transistor Q1, with its body oriented as shown. You may need to crank the leads out to fit the PCB pads. Solder it in place, ensuring it is pushed down firmly against the PCB. If you plan to fit another shield above this one, then its top should not be more than 10mm above the PCB.

The electrolytic capacitor should be mounted on its side to allow another board to be stacked above this one. Its longer, positive lead must go in the pad towards the top of the board, as shown.

Fit OPTO1 next. Check that its notch or pin 1 dot faces in the direction shown. Carefully bend the pins to allow it to fit into the PCB pads and solder it in place.

Headers

The various headers should be fitted next. Note that if you already know which Arduino pins will be used for the DIR, ENABLE and ISENSE signals and they will not change, you could omit JP1-JP3 and fit wire links in their places.

To connect to the Arduino, you can use either regular headers or stackable headers. We recommend using the Arduino board as a jig to ensure that the pins are square and flush to the PCB.

Stackable headers can be more tricky to mount as they need to be soldered from below. If possible, use those with 11mm-long pins (some that have 8mm pins don't leave much room to solder).

Thread the headers through the shield and into the Arduino board. Flip the whole assembly over so that the shield is resting flat against the pins, then solder the end pins of each group in place to secure the headers.

You can then remove the shield from the Arduino board and solder the remaining pins in place, before retouching the end pins.

It's easiest to use single-row pin headers for JP1-JP3 – snapped to length and soldered side-by-side for JP1 and JP2.

If you are snapping 40-way headers to do this, you will need at least two. Rather than fitting JP3 as a separate two-way header, you can make the top two rows of JP1 longer by one pin (ie, 15 pins rather than 14).

The last step in the construction is to fit the two screw terminals to CON1 and CON2, with their wire entry holes facing the outside edge of the board. Ensure that they are flat against the PCB; this is particularly important if you need to stack a shield above this one.

You may need to trim the underside of CON2, as this could foul the DC jack of an attached Uno board. Similarly, the underside of CON1 comes close to the metal shell of the USB connector of an attached Uno.

It's a good idea to add a layer of electrical tape on top of the USB connector on the Arduino board, to make sure they can't short if the boards flex.

Jumper settings

We suggest that you connect DIR to D10, ENABLE to D3 and ISENSE to A0, as shown in Fig.2 and Fig.3. This suits our software. Note that there are triangular silkscreen markings on the PCB to indicate the default jumper locations for JP1. To use the board as a *DCC Booster* with our supplied software, just add a fourth jumper across JP3 at upper-right.

Software

There are a few different ways this shield can be used, and each has its own software requirement. We'll describe a few of these possibilities. The following assumes that you have fitted the jumpers to the default locations described above.

DCC++

We mentioned the DCC++ software in our October 2019 article. It is designed to work with either an Uno or Mega board; we paired it with the Uno previously, and the discussion in this article assumes the same.

The Uno is adequate to work with the JMRI (Java Model Railroad Interface) software and will naturally cost less than a Mega.

The DCC++ project also includes a Processing-based GUI application for your PC that can interface with the *Base Station*, although this has been customised to work with a layout which belongs to the DCC++ software designer.

Alternatively, you can use JMRI. We also covered this software in the previous article. JMRI can be downloaded from **http://bit.ly/pe-jan21-dccb**

There are versions for macOS, Windows and Linux. It can even be run on Raspberry Pi single-board computers.

Carefully follow the installation instructions, including installing Java if necessary.

As we mentioned, our hardware is compatible with DCC++ in base station mode.

There is more information, including the required Arduino sketch, available for download from: **http://bit.ly/pe-jan21-dccc**

Parts list – Arduino DCC Controller

1 Arduino Uno or equivalent
1 12-22V DC high-current supply (see text)
1 double-sided PCB coded 09207181, 68.5mm x 55mm, from the *PE PCB Service*
1 set of Arduino headers, standard male or stackable (1 x 6-way, 2 x 8-way, 1 x 10-way)
2 2-way 5/5.08mm pitch PCB-mount screw terminals (CON1,CON2)
[Jaycar HM3172, Altronics P2032B]
2 15-way pin headers (JP1,JP3)
1 14-way pin header (JP1)
2 6-way pin headers (JP2)
4 jumper shunts/shorting blocks

Semiconductors
2 BTN8962TA half-bridge drivers, TO263-7 (IC1,IC2) [Digi-key, Mouser]
1 6N137 high-speed optoisolator, DIP-8 (OPTO1)
1 BC549 100mA NPN transistor (Q1)
1 green 3mm LED (LED1)
1 red 3mm LED (LED2)
1 1W or 5W zener diode to suit your situation (ZD1; see text)
3 1N4148 signal diodes (D1-D3)

Capacitors
1 100µF 35V electrolytic
2 100nF MKT

Resistors (all 1/4W 1% metal film)
1 100kΩ 1 20kΩ 3 10kΩ 1 2.2kΩ 5 1kΩ 1 330Ω

This software is designed to work with several commonly available Arduino motor driver shields. But these shields need some modifications to work, whereas our hardware only requires the correct jumpers to be set. The default setting in DCC++ for the MOTOR_SHIELD_TYPE of '0' will work with our hardware.

Open the Arduino IDE, select the Uno board and its serial port via the menus and open the DCC++ *Base Station* sketch that you've downloaded. Then upload the sketch to the Uno. If you open the serial monitor at 115,200 baud, you will see a banner message; this indicates that the *Base Station* software is working as expected.

You can also interact with the *Base Station* through serial commands. The protocol is detailed in the PDF file that is included in the DCC++ *Base Station* project ZIP file.

Once you've tested this, close the Serial monitor and open the JMRI DecoderPro program. Go to Edit -> Preferences, and under Connections, choose DCC++ as System Manufacturer, DCC++ Serial Port as System connection. Ensure the serial port setting matches the Uno's.

Save the settings and close DecoderPro, so that it can reload the new settings. Re-open DecoderPro and under Edit -> Preferences, choose Defaults, and ensure that the name of the new connection name is used for all connections (instead of 'Internal').

Unless you have other hardware you want to use, you should select DCC++ for all options.

Save, close and re-open DecoderPro again. Click the red power button in DecoderPro and ensure that the power button turns green. The LED on the DCC *Power Shield* should switch from red to green.

The simplest way to drive trains is to select Actions -> New Throttle, set the locomotive address and manipulate the controls (see Screen1).

JMRI can do a lot of different things, so we suggest you read its manual to find out about its capabilities. The JMRI project also includes PanelPro, which can be used to design track and signal diagrams for controlling a model layout.

Adding the DCC Programmer

If you have already built the *DCC Programmer*, then the Arduino board is already programmed to work with the *DCC Power Shield*, and the *DCC Power Shield* can be added to the stack, ideally at the top.

As we noted earlier, the choice of zener diode and power supply will be more complicated if you want to construct an all-in-one setup. Since this is likely to be a smaller system, we suggest that a modest power supply will be suitable.

Using the DCC++ software with JMRI is the same as discussed above.

Using it as a booster

When a signal is fed in via the optoisolated input (CON3), the *DCC Power Shield* is effectively working as a *Booster*. The signal can be from another *Base Station* or system, with the *DCC Power Shield* turning that signal into a more powerful DCC signal that can be used to drive trains.

While it might not seem that an Arduino is needed in this case, it's a good idea to have one as we can program it to monitor the DCC signal and intervene if there is a problem. So we've written a sketch to allow an Arduino to take on this supervisory role.

There are two main conditions to check for. First, we want the *Booster* to be able to protect the shield if too much current is being drawn from it.

This could be due to an overload or even a short circuit, such as a metal object being dropped across the tracks.

Thus, our sketch continually monitors the voltage present on its A0 pin via its internal analogue-to-digital converter (ADC). If it gets above a certain threshold, the power to the track is cut by pulling the ENABLE pin low.

A timer starts and the sketch attempts to re-apply power after it expires. If the short circuit is still present, then the over-current condition re-occurs, power is cut again and the timer re-starts.

The other condition we need to consider is if the incoming DCC signal is lost. This could be for any reason, such as if the connection to CON3 is broken or the upstream *DCC Base Station* has a fault. In any case, when there is no signal at CON3, the input to IC1 is held high and IC2's input is low. There is then an unchanging DC voltage across the tracks.

This may not sound like a problem, but some DCC locomotives can be programmed to undergo 'DC conversion'.

When a locomotive decoder detects that there is a steady DC voltage present, the locomotive behaves as if it was on a conventional 'single-throttle' layout and will typically set off in one direction at full speed (hopefully not towards the end of the track…).

This feature was initially added to allow DCC locomotives to be used on conventional layouts, perhaps as an aid to owners transitioning to DCC from DC systems.

Fortunately, the DC conversion feature can be turned off in the decoder by setting a configuration variable. You can use a *DCC Programmer* such as from our October 2019 article to do this.

In any case, the sketch detects that the DCC signal is no longer changing and pulls the ENABLE line low, disabling the track output and preventing such runaways.

To enable the use of the optoisolated input, add a jumper across JP3. Leave the jumper on 'DIR' for pin D10 in place; D10 is set as an input in the software and is used to monitor the incoming DCC signal.

The *Booster* sketch is called **DCC_Shield_passthrough_supervisor.ino**. This uses a library to perform the precision timing needed to generate the DCC waveform, called **TimerOne**.

This can be installed via the Library Manager by searching for 'timerone' or from the ZIP file we have included with our software package.

Open the sketch, select the Uno and the serial port and upload it. Disconnect the USB cable and connect your power source to CON1. The red LED should light. Connect a valid DCC signal to CON3 and the green LED should light. You should then have a valid DCC signal at CON2.

A standalone sketch

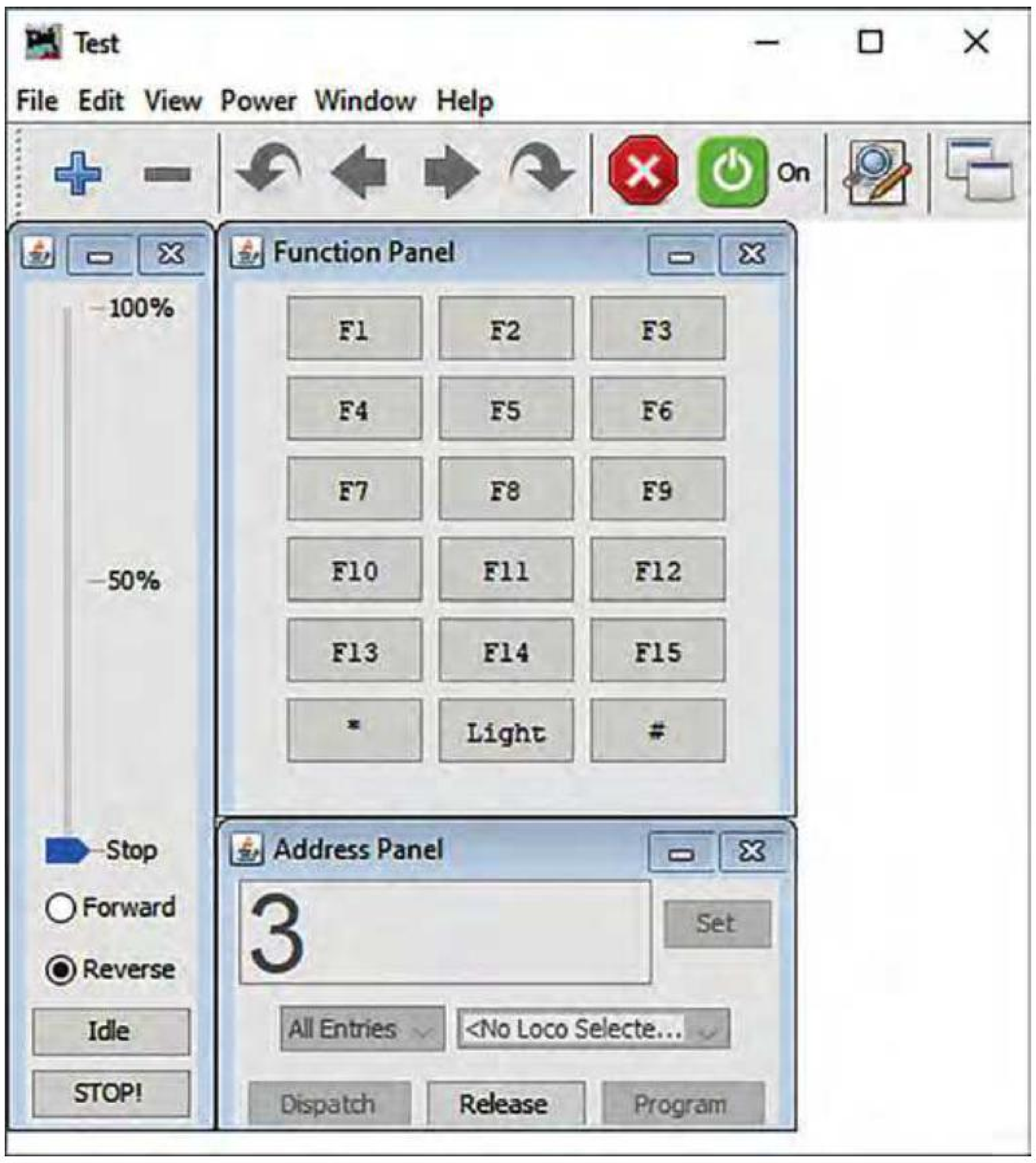
We've also created a simple standalone sketch that produces a DCC signal, suitable for controlling a single locomotive.

The decoder identification number has been set to 3 (which is the default for new, unprogrammed decoders), although it can be changed in the code. We suggest you use this option if you want to try out DCC for the first time.

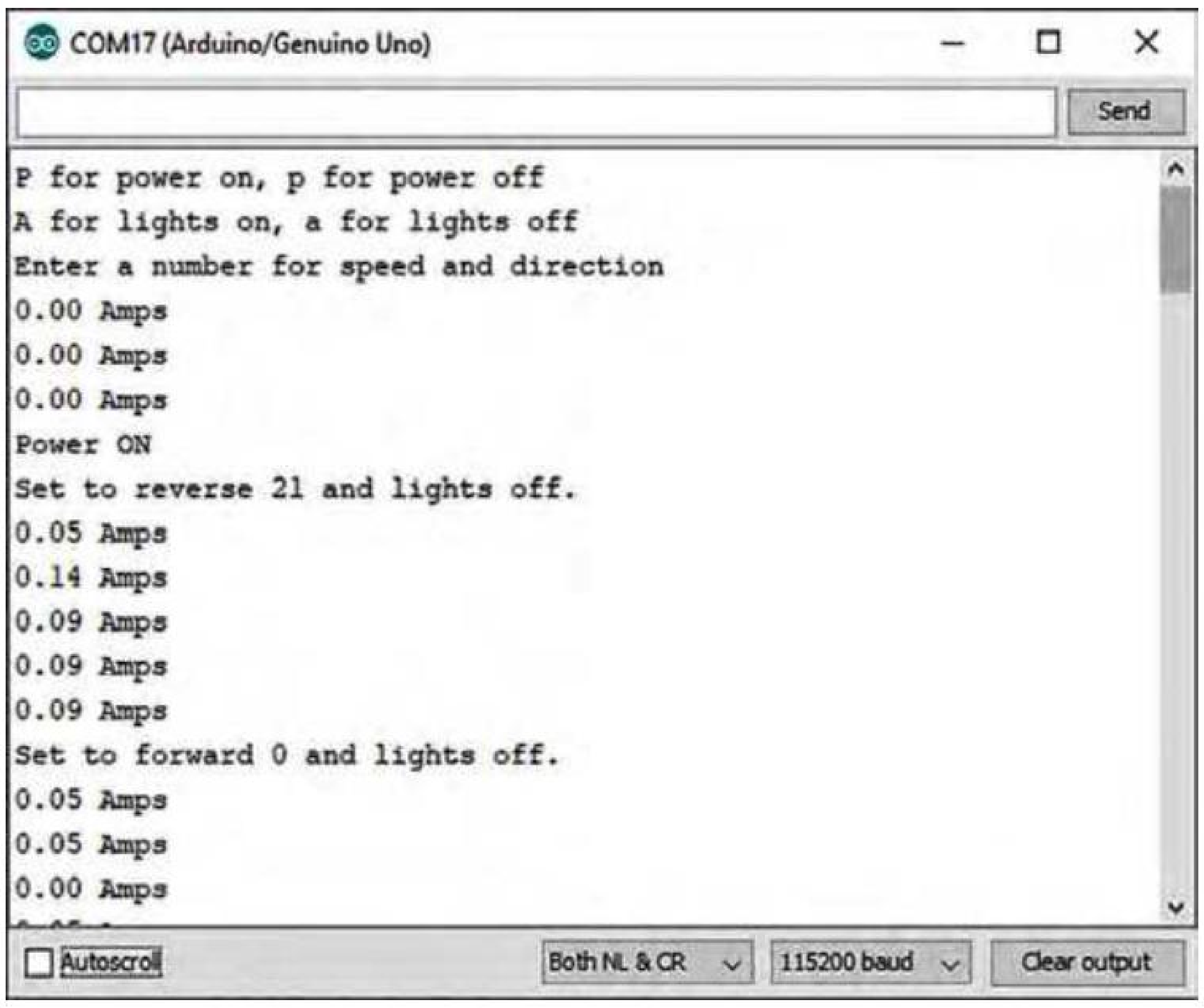
We can't offer advice on fitting decoders; there are so many options for both decoder choices and how they are connected.

The companies that manufacture the decoders do offer advice (and many have custom decoders to suit specific locomotives). After all, they want to make it easy for you to buy their products.

Our standalone sketch also requires the **Timer One** library mentioned above, so make sure that is installed.



Screen1: while JMRI's DecoderPro program has many features, it also has a set of basic tools for controlling trains. This throttle window allows speed, direction and light functions to be controlled. You can even switch track power directly; the green icon at upper right mimics the status LEDs on the shield.



```
P for power on, p for power off
A for lights on, a for lights off
Enter a number for speed and direction
0.00 Amps
0.00 Amps
0.00 Amps
Power ON
Set to reverse 21 and lights off.
0.05 Amps
0.14 Amps
0.09 Amps
0.09 Amps
0.09 Amps
Set to forward 0 and lights off.
0.05 Amps
0.05 Amps
0.00 Amps
```

Screen2: while very basic, our standalone sketch named DCC_Single_Loco_Control.ino **allows power, speed, direction and lights to be controlled by commands in the serial monitor – download from the January 2021 page of the *PE* website. The software can be modified to control multiple locos. Advanced Arduino users could use it as the basis of an automated layout control system.**

Set the jumpers on the shield to the default positions and connect the Uno to the computer. Open the **DCC_Single_Loco_Control.ino** sketch and select the Uno board and its serial port. Press the Upload button to compile and upload the sketch, then open the Serial Monitor at 115,200 baud (see Screen2).

You can now enter commands as numbers which correspond to the desired locomotive speed, in 128 steps. Thus, numbers from –127 to 127 are accepted. You should ensure that 28/128 step speed mode is set on your locomotive decoder.

Type 'P' (upper case) to turn track power on and 'p' (lower case) to turn it off. The power will automatically turn off if current over half an amp is detected. You can also use 'A' and 'a' to turn on and off the loco's headlights.

The program is elementary, but it has several unused functions to send all manner of DCC packets to the track. If you are comfortable with the Arduino, you should have no trouble adapting it to do something more advanced.

Current limitations

Using the specified components and the DCC++ software, the shield can easily deliver up to 10A. This is mostly limited by the screw terminal connectors. The DCC++ software also has a hard-coded current limit which kicks in at around 10A.

Of course, the software limit is easy to change, but any hardware changes should be done with care.

The output driver ICs are capable of handling around 30A, with the PCB tracks topping out around 20A.

In any case, everything runs cool well below the 10A limit, so maintaining this limit is good for component longevity.

DCC has a wide range of operating voltages, so to increase output power, it may be easier to increase the supply voltage.

Most locomotives use PWM speed control on their motors, so a higher supply voltage simply means a lower PWM duty cycle (and thus current consumption) for the same speed.

We haven't done any tests above 10A, but if you are set on increasing the current capacity of the *DCC Power Shield*, then you should ditch the screw terminal connectors and solder thick copper wires directly to the board (ideally, to the power pins of IC1 and IC2).

If the wires can handle 20A, then your modified *DCC Power Shield* should have no trouble doing that.

To go higher than this will probably mean that IC1 and IC2 need some heatsinking, as well as even thicker wires. We suggest that you instead consider using more, smaller boosters. For example, you could modify the *Booster* sketch to monitor and drive multiple *DCC Power Shields* stacked above it.

A larger system

If you are planning a system with multiple *Boosters*, either because you need the power or it otherwise makes sense to do so, then there are a few minor caveats.

When running multiple *Boosters*, avoid daisy-chaining the DCC signal from one *Boosters* to the next. Instead, fan out the DCC signal from one *Base Station* to all the *Boosters*.

Many commercial *Base Stations* have a low-powered DCC signal output (Digitrax names this Railsync), which is ideally suited for this purpose.

The first problem with a daisy-chain configuration is that if one *Booster* goes down, then so do all those downstream, as the DCC signal will be shut off.

Second, each *Booster* also has a small but measurable delay in propagating the signal. In our case, this is around 4µs, due to the switching time of the BTN8962s.

This delay is not usually a problem, but it may become one at the boundary where the tracks from two *Boosters* meet (where there would typically be an insulator, to prevent one *Booster* feeding another *Booster's* section of track).

Where the tracks meet, a train may be briefly fed by both the *Boosters*. If there is a delay between the signals from the two *Boosters*, then it may appear to be a short circuit if the two *Boosters* are delivering opposite polarity voltages at that instant.

This is less likely to occur if the *Boosters* are well synchronised, which should be the case if all are being fed the same signal.

You should also ensure that the *Boosters* are fed with similar supply voltages, so that one *Booster* does not try to power another *Booster's* track when the train bridges their join.

You must also ensure that the *Boosters* are wired with the correct polarity where the tracks meet.

Using the DCC Booster Shield as a motor driver

The *DCC Booster Shield* can be used as a high-current motor driver shield. In this case, the signal on the DIR pin determines the motor direction, and a pulse-width modulated signal is applied to ENABLE to control the speed.

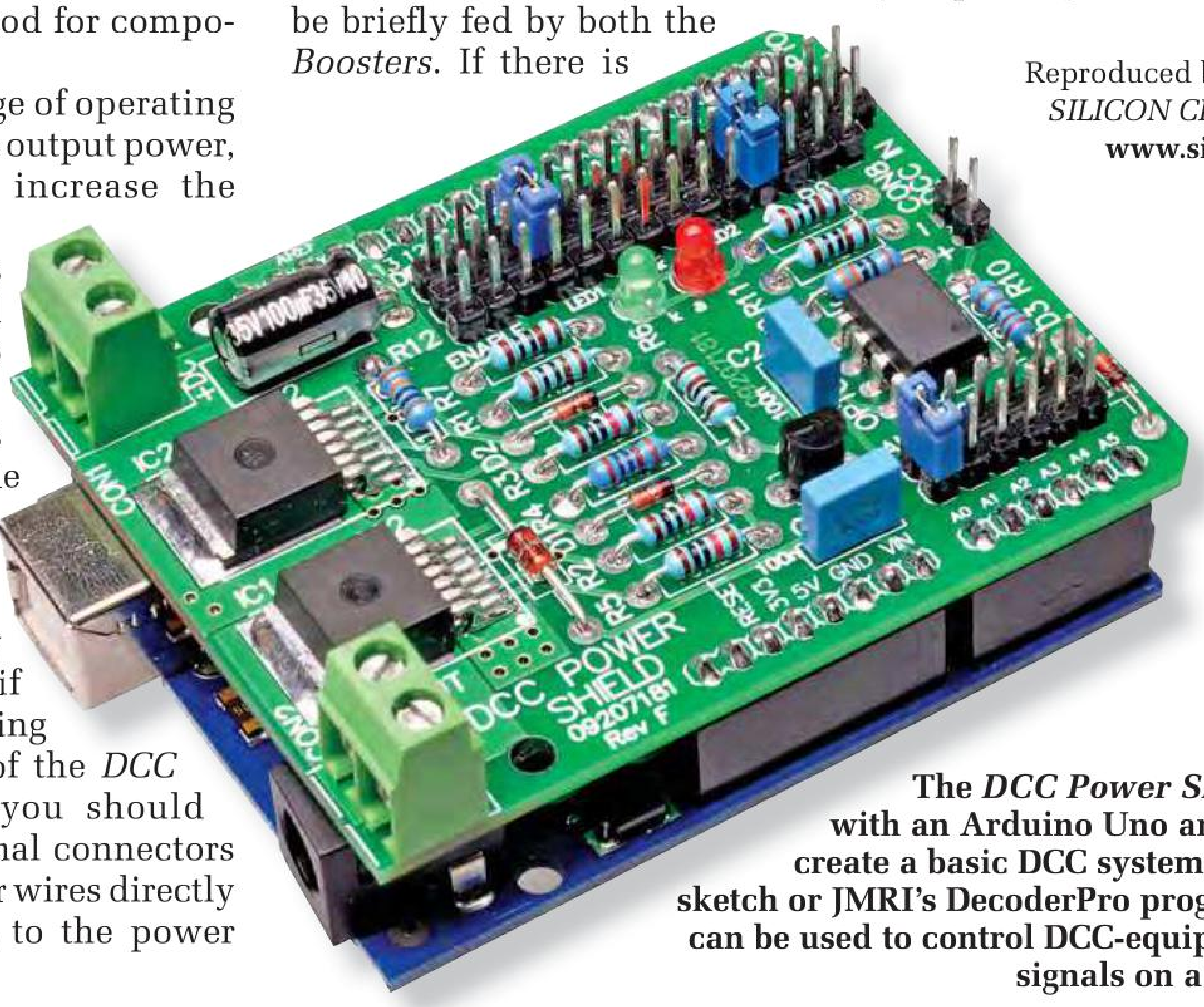
The BTN8962 has active freewheeling, so no external diodes are needed. If used like this, LED1 and LED2 will both appear to be on at the same time, with green LED1 becoming brighter and red LED2 dimmer as the duty cycle increases.

As noted earlier, the 100µF electrolytic capacitor is adequate for a DCC application. A larger value may be needed for motor driving.

We suggest leaving ZD1 off, as larger motors will create hefty spikes at the end of each drive pulse.

Keeping the two supply rails separate will prevent this from damaging the Arduino board.

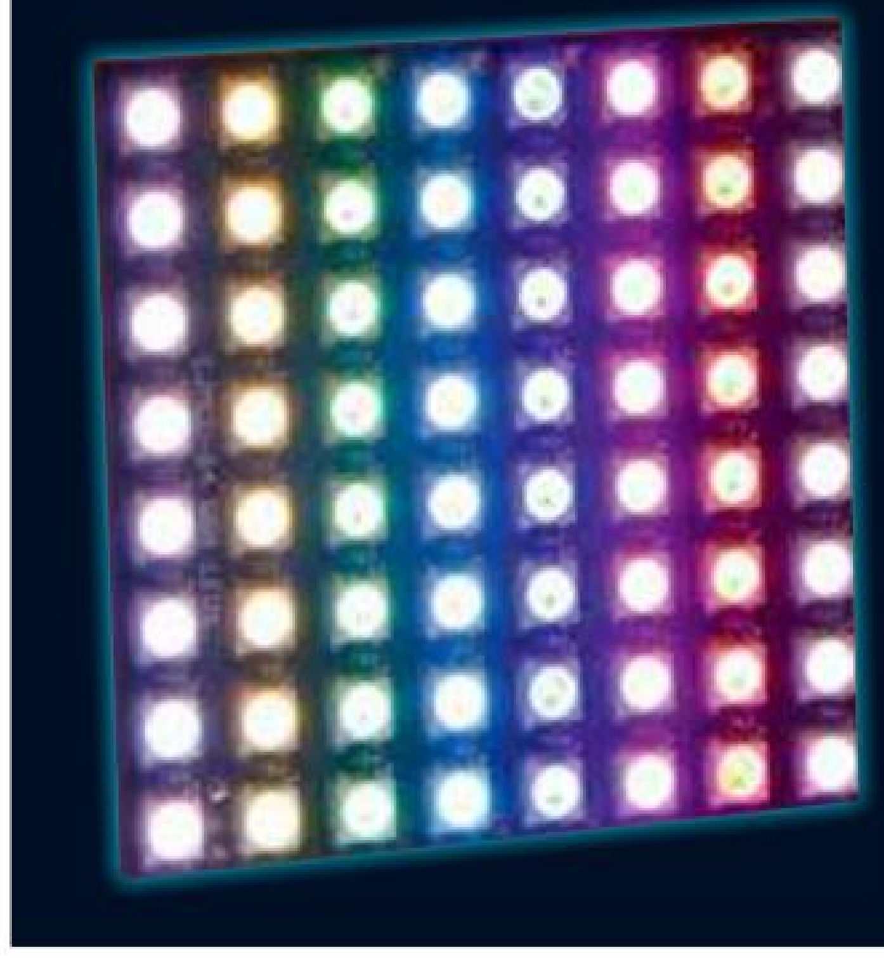
Reproduced by arrangement with
SILICON CHIP magazine 2020.
www.siliconchip.com.au



The *DCC Power Shield* can be combined with an Arduino Uno and DC power supply to create a basic DCC system. Using our standalone sketch or JMRI's DecoderPro program, this combination can be used to control DCC-equipped trains, points and signals on a model railway layout.

Using Cheap Asian Electronic Modules by Jim Rowe

Intelligent 8x8 RGB LED matrix



This month, we're looking at a module with an 8x8 matrix of 64 'intelligent' RGB LEDs. Each LED can display over 16 million different colours, or primary colours at 256 brightness levels. The LEDs are controlled serially via a single wire, and multiple modules can be cascaded to build a much larger display, enabling all sorts of useful applications.

We looked at some 8x8 LED display modules in an earlier article in this series, back in the July 2018 issue. We thought it was worth writing this one up too, as it is significantly more flexible and just generally more useful.

It uses RGB (red/green/blue) LEDs rather than monochrome (single colour) LEDs. Each LED can display up to 256 brightness levels for each of the three colours, to give a total of 16,777,216 (256 × 256 × 256) different colours.

In this module, each RGB LED has its own built-in serial data register, latch register and decoder/driver, so no separate controller is needed.

All 64 LEDs of the module are connected in sequential (daisy-chain) fashion, so that serial data can be fed into the first LED of the module and passed through to the other LEDs in turn.

If you want to use multiple modules, the data output from the 64th LED on the first module can be fed to the first LED of the next module to program its LEDs as well. And so on.

This module is based on an impressive device: the WS2812B intelligent control LED made by WorldSemi, based in Dongguan, Guangdong province, China (between Guangzhou and Shenzhen, and near Hong Kong).

I should note that some of the modules currently available use a 'clone' of the WS2812B device, the SK6812, made by another Chinese firm: Shenzhen Sikewei Electronics. Although the timing specs for the SK6812 differ a little from those of the WS2812B, they are quite compatible with most of the available software.

You can find these WS2812B/SK6812-based 8×8 RGB LED modules on the internet from various vendors, many of them available via sites like eBay or AliExpress (**www.aliexpress.com/item/32671025605.html**). The prices vary quite a bit, but you can find them from around US$4 shipped!

Now let's look at the WS2812B IC to see how it works. (This description also applies to the SK6812.)

The WS2812B LED chip

This small (5 × 5 × 1.6mm) four-lead SMD package, shown in Fig.1, houses a trio of LEDs as well as a serial controller IC. It looks deceptively simple, but you can see from the block diagram (Fig.2), there's quite a lot inside.

It includes a 24-bit shift register, a 24-bit latch, three eight-bit DACs (digital-to-analogue converters) coupled to a driver for each LED and a buffer amplifier to boost and reshape the serial data output, ready for the next WS2812B.

Fig.3 shows how a string of 64 WS2812B devices are connected to

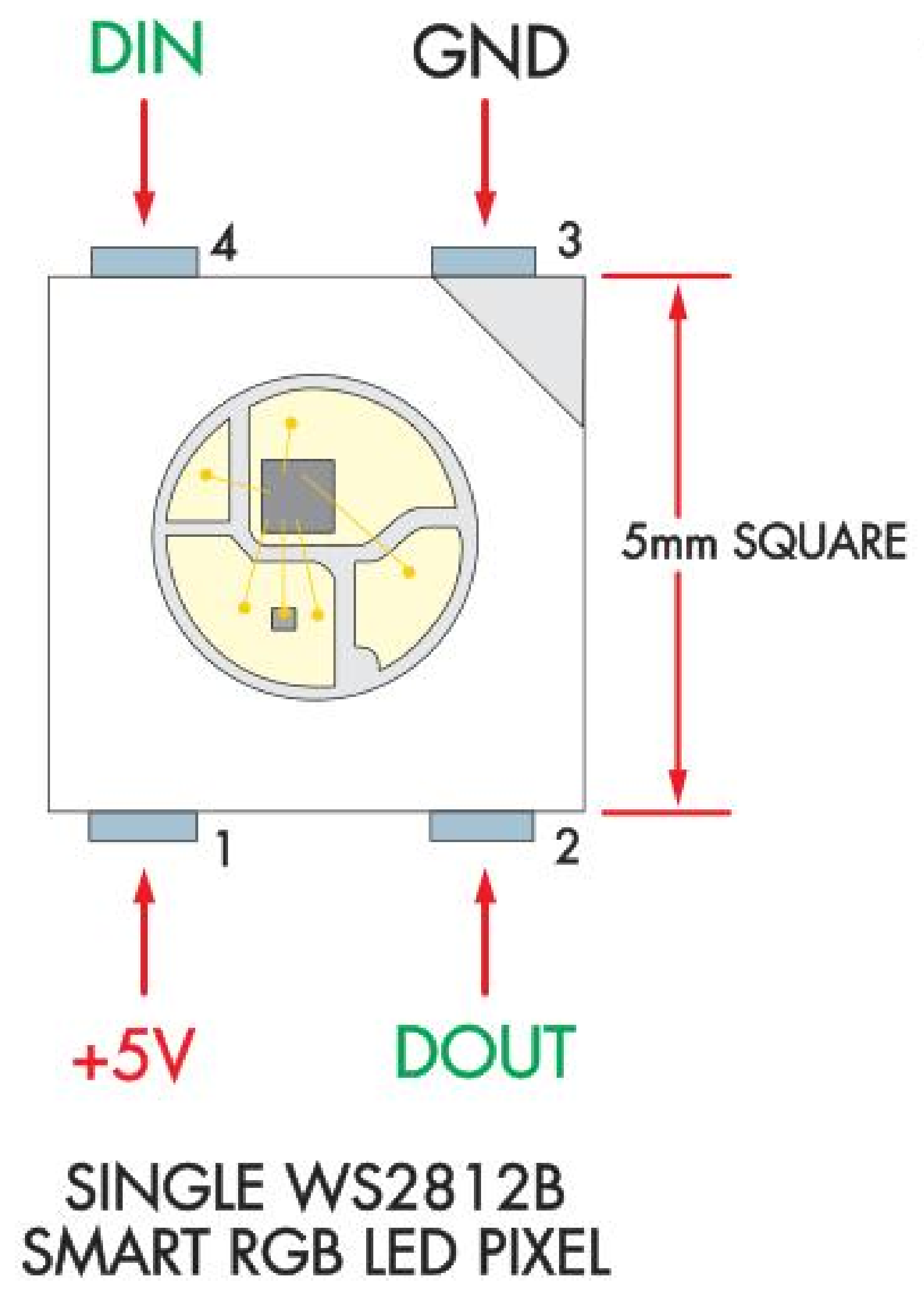


Fig.1: the SMD package size and pinout of the WS2812B (and equivalent) chips. Internally, it's made from multiple semiconductor dies, tied together with bond wires and encapsulated with a plastic lens on top. Note that the package orientation marking is located on pin 3, rather than pin 1.

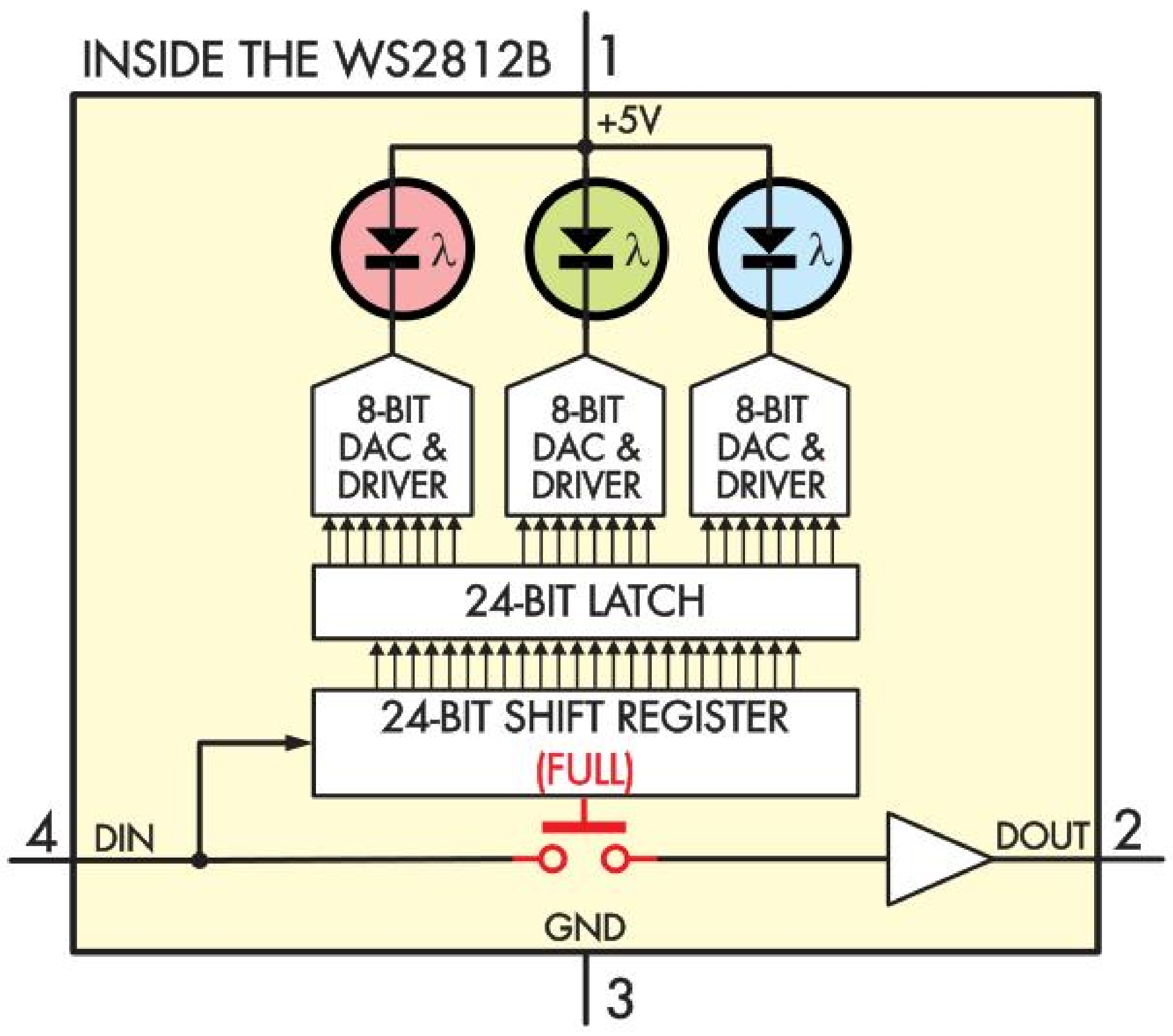


Fig.2: as well as the red, green and blue LED dies, the WS2812B incorporates a controller/driver IC, which includes a serial latch plus three linear LED drivers with 8-bit DACs.

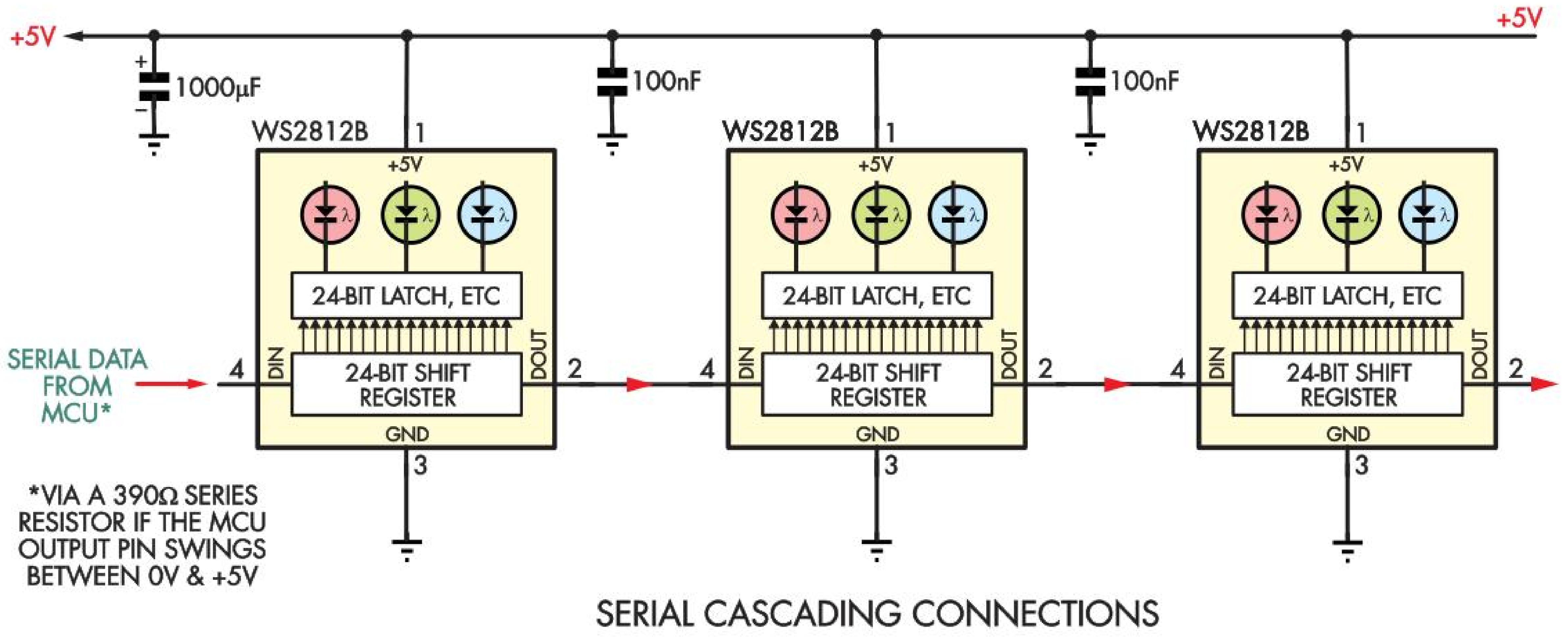


Fig.3: cascading multiple WS2812B devices is simple. The DOUT (data out) pin of one device is simply connected to the DIN (data in) pin of the next device. The 5V and GND pins are all connected in parallel, with a 100nF bypass capacitor close to each device.

make up the module. This is simplified by showing just three of the 64 devices. The data stream from the MCU is fed into pin 4 (DIN) of the first device, while the output from pin 2 (DOUT) is connected to pin 4 of the next device, and so on.

One of the slightly unusual features of this chip is that unlike other daisy-chained shift registers, it doesn't feed the top-most 'overflow' bit of the shift register to the output, for feeding into the next device.

Rather, the output is held in a static state until all 24 bits have been shifted into the register (presumably, tracked via a counter register), at which point it no longer shifts in any new bits. The input is then connected to the output buffer via an internal switch.

This means that the first 24 bits of data shifted into the daisy chain determine the state of the first device. With the more typical (and simpler) shift-through design, the first bits of data end up in the last device – ie, you have to shift in the data in reverse order.

So, presumably the reason for this unusual scheme is to avoid the need to reverse the order of data being sent to an array of these devices.

The only other components are the 100nF bypass capacitors on the +5V supply line, with one next to each device. The 1000µF reservoir capacitor is external to the module.

The physical layout of the 64-LED array, which measures 65 × 65mm, is shown in Fig.4. The input connections for the module are at lower left, while the output connections are at upper right.

Each WS2812B device can draw up to 18mA from the +5V supply during operation, so a single 64-LED module can draw as much as 1.152A.

That's why it's recommended that even using a single module, the +5V supply for the module should not come from your MCU (eg, Arduino or Micromite), but from a separate DC supply.

It's even more important to do this when you're using several modules in cascade. This is also why that 1000µF capacitor is needed on the +5V supply line.

Driving the module

The LEDs in these modules are programmed serially via a single wire, as mentioned earlier. But they use a special pulse-width modulation (PWM) coding system for the data, shown in Fig.5.

The timing for a zero bit, a one bit and the RESET/LATCH pulse for a basic WS2812B device are shown at the top of Fig.5; this is used in most of the currently-available 8×8 modules. The corresponding timings for the latest WS2812B-V4 version of the device are shown adjacent.

There are subtle differences in data bit timing between the two versions. The main difference is that the WS2812B needs a RESET/LATCH pulse lasting more than 50µs, while the WS2812B-V4 needs a longer pulse of more than 280µs.

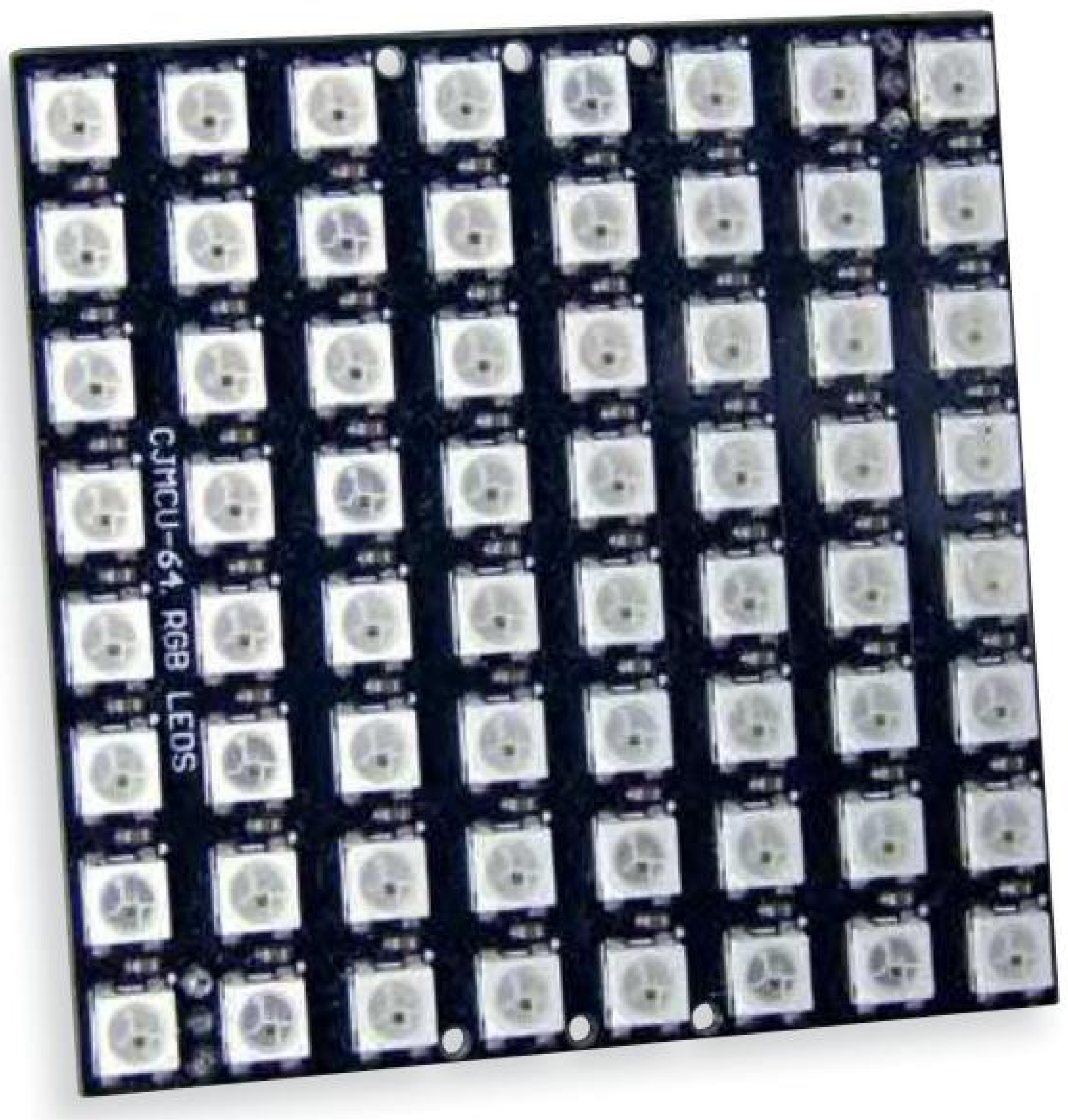
Timing for the SK6812 device is similar to that for the WS2812B, with a zero bit composed of a 300ns high followed by a 900ns low, a one bit composed of a 600ns high followed by a 600ns low, and the RESET/LATCH pulse needing to be 80µs or more.

The centre section of Fig.5 shows the 24-bit data packet used to program a single WS2812B LED. There are eight bits for each of the three colours, with each colour's data byte sent MSB (most-significant-bit) first. So the total time needed to refresh one LED is either 30µs or 26.4µs, depending on the version of the WS2812B chip.

Fig.5 also shows the colour data sent in GRB (green-red-blue) order, but some of the WS2812B or equivalent devices used in these modules require the data to be sent in RGB order. As a result, much of the software written for these modules allow the colour byte order to be changed to suit the specific devices being used.

The 64-LED data stream used to program all of the WS2812B LEDs in a single 8×8 module is shown at the bottom of Fig.5. As you can see, the 24 bits of data for each of the 64 LEDs are sent in turn, followed by a RESET/LATCH pulse. This pulse instructs all of the WS-2812Bs to transfer the data in their shift register into the latch register, changing the colour and brightness of its LEDs to the new values.

So one complete refresh cycle for an 8×8 module takes very close to 1970µs



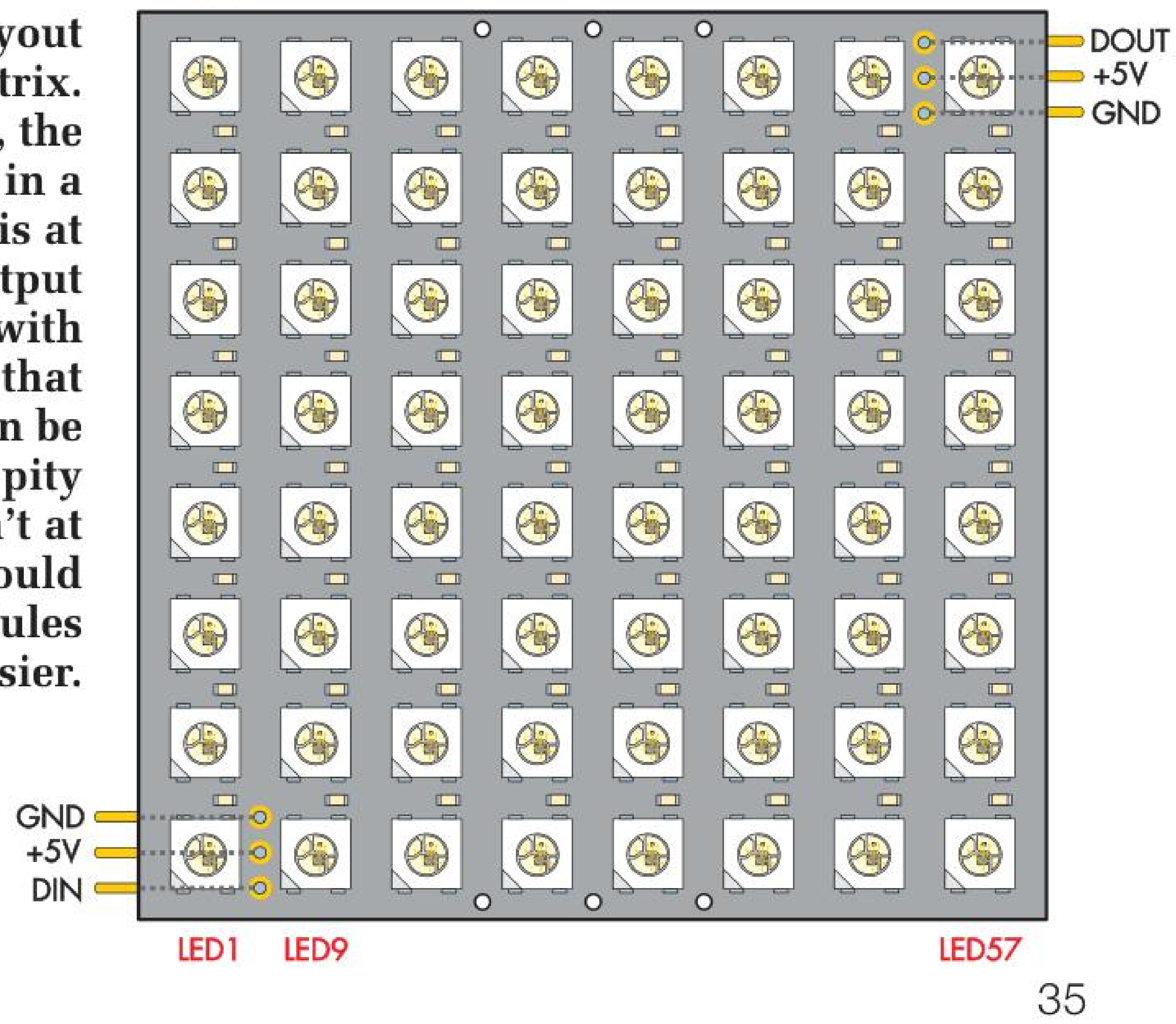
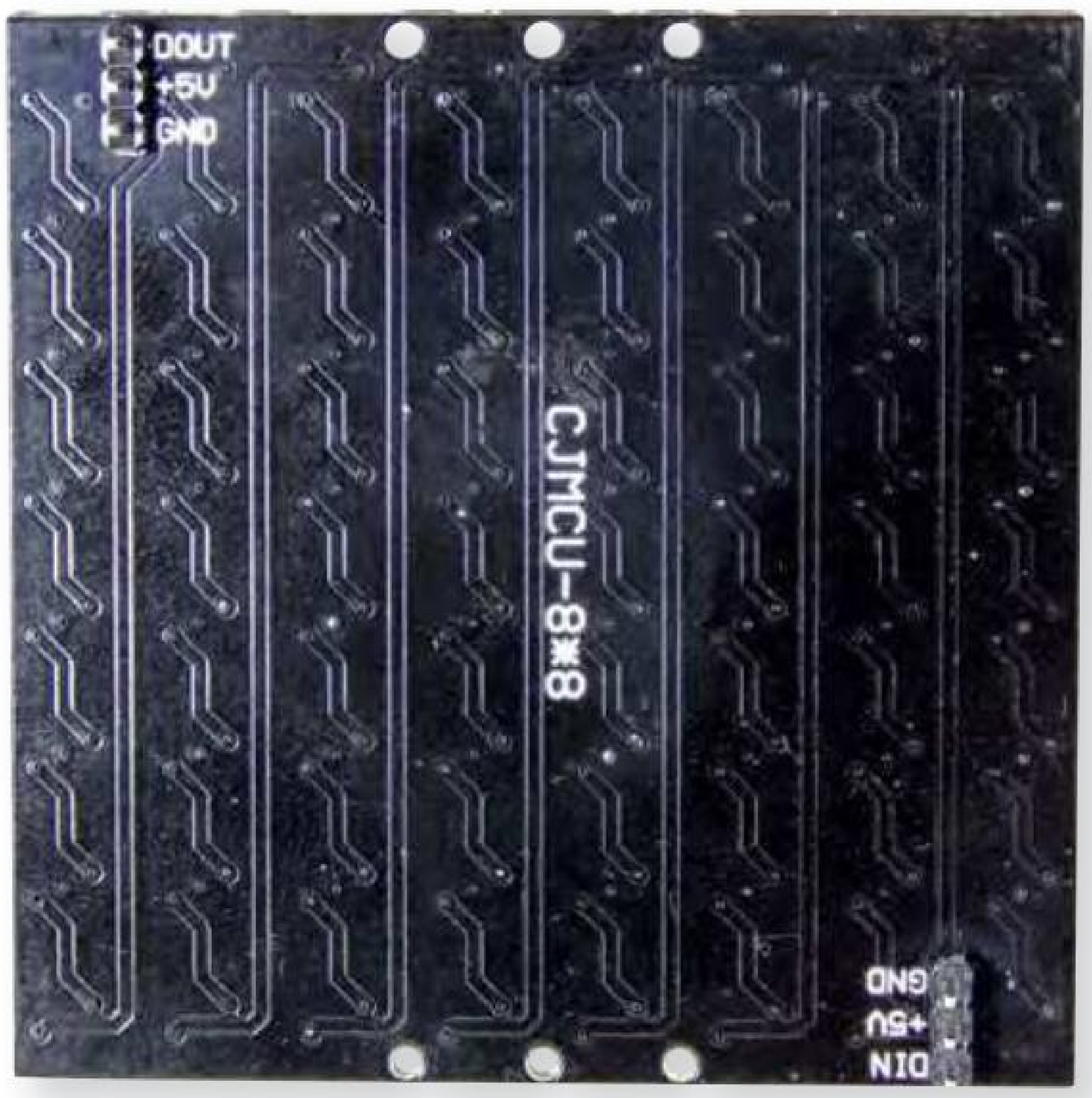


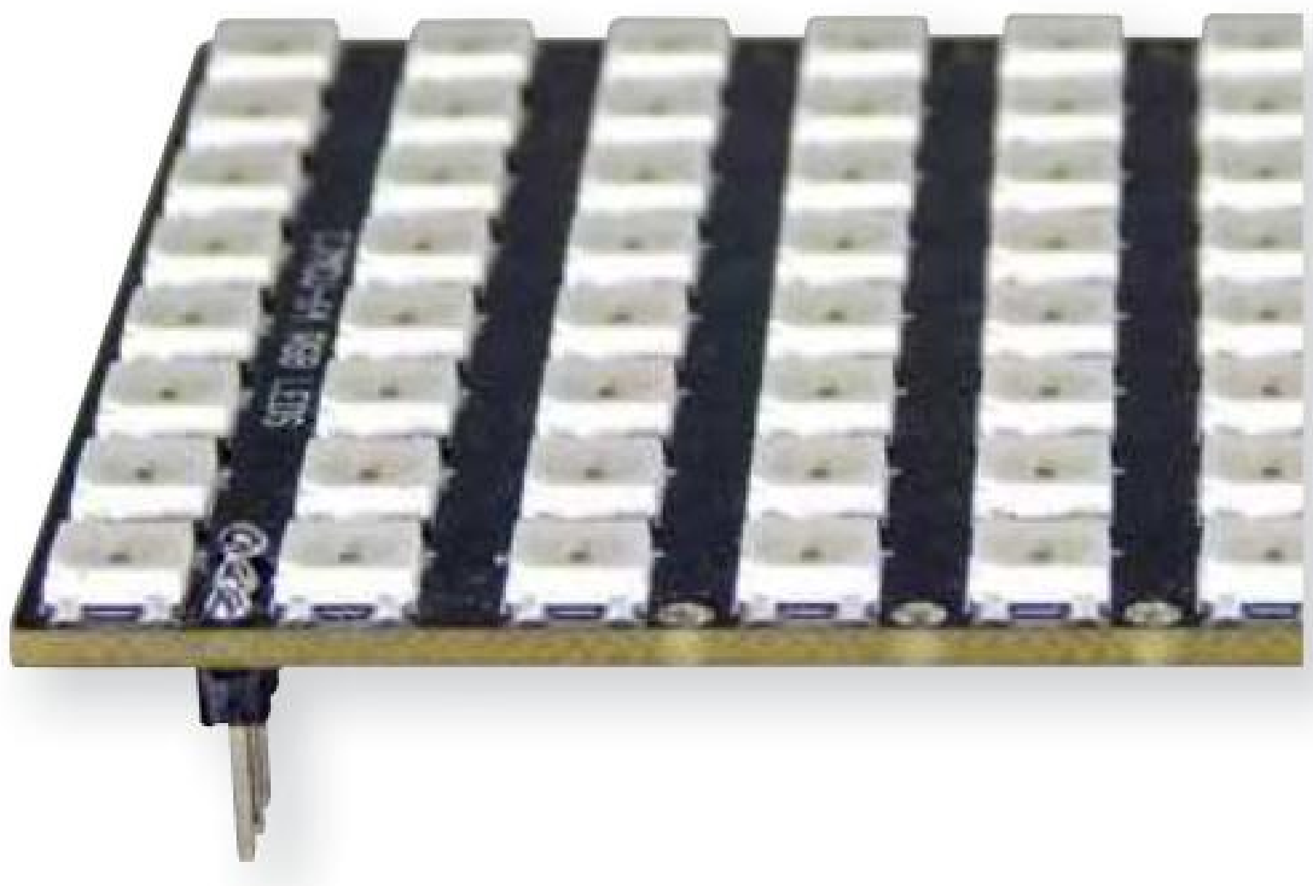
Fig.4: this shows the layout of the 8×8 RGB LED matrix. As you would expect, the LEDs are laid out in a grid. The data input is at lower left and data output at upper right (along with the supply pins), so that multiple modules can be daisy-chained. It's a pity that the output isn't at lower right, as that would make chaining modules considerably easier.



DOUT
+5V
GND
CJMCU-8*8
GND
+5V
DIN

While the underside of this module uses headers for external connections, some modules provide SMD pads rather than holes. It can be worthwhile to shop around, but there is a risk that you may come across clones which are not fully compatible.

This 8×8 RGB LED module uses WS2812B ICs. The data and power connections are made via two 3-pin male headers on the PCB's underside.



(1.970ms) or 1969.6µs (1.969ms), depending on which version of the WS2812B is being used. As a result, the display can be refreshed up to 500 times each second (or a fraction of this with multiple modules, eg, 100 times per second for five modules daisy-chained).

Driving it from an Arduino

Thanks to the single-wire data programming system used by the WS2812B device, it's physically quite easy to drive this module from an Arduino.

As shown in Fig.6, all that's needed is a wire connecting the module's GND pin to one of the Arduino GND pins, together with a wire with a 390Ω series resistor connecting the module's DIN pin to one of the Arduino's digital I/O pins.

Wires from the module's +5V and GND pins are then used to supply it with 5V power, with a 1000µF capacitor used as a reservoir to ensure that the 5V power remains constant.

Writing the required Arduino 'sketch' (program) is a little complicated due to the unusual PWM coding system used. Luckily, several Arduino software libraries have been written to drive a string of WS2812B/SK6812 devices.

You'll find suitable programs in various places on the Web, most of them fairly simple and straightforward. Many of them make use of a library of routines for the Arduino written by the Adafruit people and called **Adafruit_NeoPixel**.

To get you started, I've written a sketch called **RGBLED_Matrix_sketch.ino**, available for download from the January 2021 page of the *PE* website. It uses the **Adafruit_NeoPixel** library, which can be downloaded from **https://github.com/adafruit/Adafruit_NeoPixel** (or via the Arduino IDE's Library Manager).

This sketch allows you to produce one of nine different patterns on the module, simply by sending a digit (from 1 to 9) to the Arduino from your PC's serial port (eg, via the IDE's Serial Monitor). For example, sending a '1' produces a changing rainbow pattern, sending a '3' produces a display of all LEDs glowing mid-green, sending a '6' produces a pattern of white dots 'chasing' each other, and so on.

While this may not sound terribly exciting, it should give you a good idea of what's involved in driving these modules from an Arduino.

Driving it from a Micromite

Driving one of the modules from a Micromite again isn't easy, mainly because of the PWM bit encoding scheme.

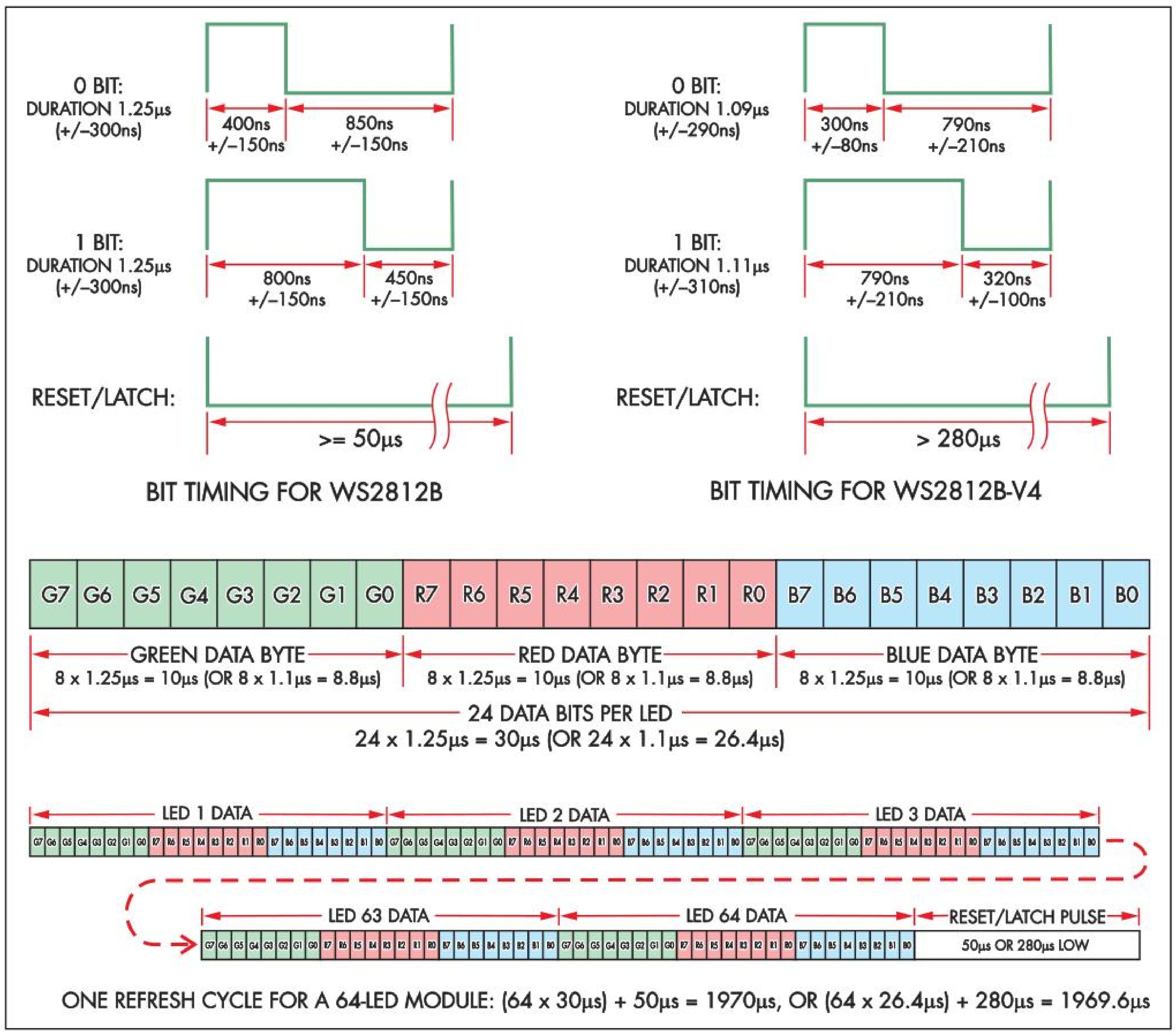
After trying to make unorthodox use of MMBasic's built-in SPI communications protocol (with no luck), I realised that I would need an embedded C function similar to Geoff Graham's SerialTX module.

CFUNCTIONs allow native 'machine language' code to be added to an MMBasic program. This would let me send the serial data streams to the LED module with the right encoding and at the right speed.

I was rather daunted at the prospect of writing this CFUNCTION. But Geoff Graham advised me that a suitable function had already been created by Peter Mather, one of the Micromite 'gurus' on *The BackShed Forum* (**http://bit.ly/pe-jan21-shed**).

I eagerly downloaded Mr Mather's CFUNCTION, and tried using it with a small MMBasic program to drive a module with 64 WS2812B LEDs. The results were a bit disappointing, with a variety of unexpected errors. This prompted me to try using my DSO to check the pulse timing of the bitstream being sent to the WS2812B LEDs, to compare it to the required timing shown in Fig.5.

I subsequently found a few differences, which seemed likely to explain the problems I was having.



0 BIT:
DURATION 1.25µs
(+/–300ns)
400ns
+/–150ns
850ns
+/–150ns
1 BIT:
DURATION 1.25µs
(+/–300ns)
800ns
+/–150ns
450ns
+/–150ns
RESET/LATCH:
>= 50µs
BIT TIMING FOR WS2812B
0 BIT:
DURATION 1.09µs
(+/–290ns)
300ns
+/–80ns
790ns
+/–210ns
1 BIT:
DURATION 1.11µs
(+/–310ns)
790ns
+/–210ns
320ns
+/–100ns
RESET/LATCH:
> 280µs
BIT TIMING FOR WS2812B-V4
G7 G6 G5 G4 G3 G2 G1 G0 R7 R6 R5 R4 R3 R2 R1 R0 B7 B6 B5 B4 B3 B2 B1 B0
GREEN DATA BYTE
8 x 1.25µs = 10µs (OR 8 x 1.1µs = 8.8µs)
RED DATA BYTE
8 x 1.25µs = 10µs (OR 8 x 1.1µs = 8.8µs)
BLUE DATA BYTE
8 x 1.25µs = 10µs (OR 8 x 1.1µs = 8.8µs)
24 DATA BITS PER LED
24 x 1.25µs = 30µs (OR 24 x 1.1µs = 26.4µs)
LED 1 DATA
LED 2 DATA
LED 3 DATA
LED 63 DATA
LED 64 DATA
RESET/LATCH PULSE
50µs OR 280µs LOW
ONE REFRESH CYCLE FOR A 64-LED MODULE: (64 x 30µs) + 50µs = 1970µs, OR (64 x 26.4µs) + 280µs = 1969.6µs

Fig.5: the WS2812B uses a custom 1-wire serial protocol, with the duration of the positive pulse distinguishing between a zero and one bit. Unfortunately, different versions of the chip require different timings, although it is possible to choose timings which will suit all versions. Note the much longer latch pulse required for the V4 chips. Also, while many chips expect colour data in the green, red, blue order shown here, some use the more standard red, green, blue order.

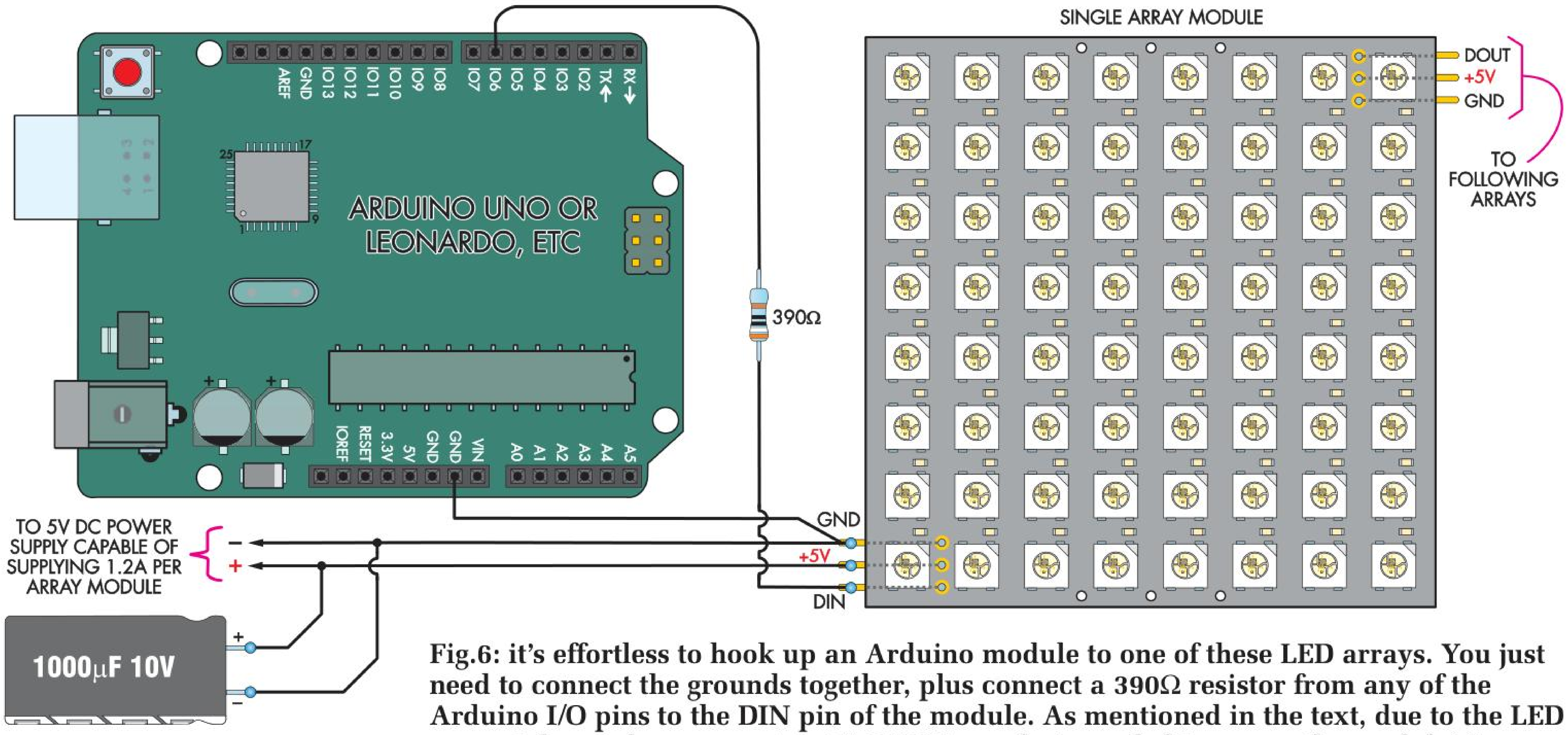

SINGLE ARRAY MODULE
DOUT
+5V
GND
TO FOLLOWING ARRAYS
ARDUINO UNO OR LEONARDO, ETC
AREF
GND
IO13
IO12
IO11
IO10
IO9
IO8
IO7
IO6
IO5
IO4
IO3
IO2
TX
RX
IOREF
RESET
3.3V
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5
390Ω
GND
+5V
DIN
TO 5V DC POWER SUPPLY CAPABLE OF SUPPLYING 1.2A PER ARRAY MODULE
–
+
1000µF 10V

Fig.6: it's effortless to hook up an Arduino module to one of these LED arrays. You just need to connect the grounds together, plus connect a 390Ω resistor from any of the Arduino I/O pins to the DIN pin of the module. As mentioned in the text, due to the LED current demands, a separate >1A 5V DC supply is needed to power the module(s).

After an exchange of emails with Mr Mather, I learned that his CFUNCTION had been written about four years ago to suit the original WS2812 LEDs.

He suggested a couple of changes to it to make the pulse timing more compatible with the WS2812B, SK6812 and WS2812B-V4 devices, and also guided me regarding how to make the changes easily without having to re-compile his code.

I made the suggested changes and tried it all again. Now the timing of the pulse stream was much closer to that needed by the WS2812B/SK6812 devices, and, lo and behold, the modules gave the correct displays from my test program.

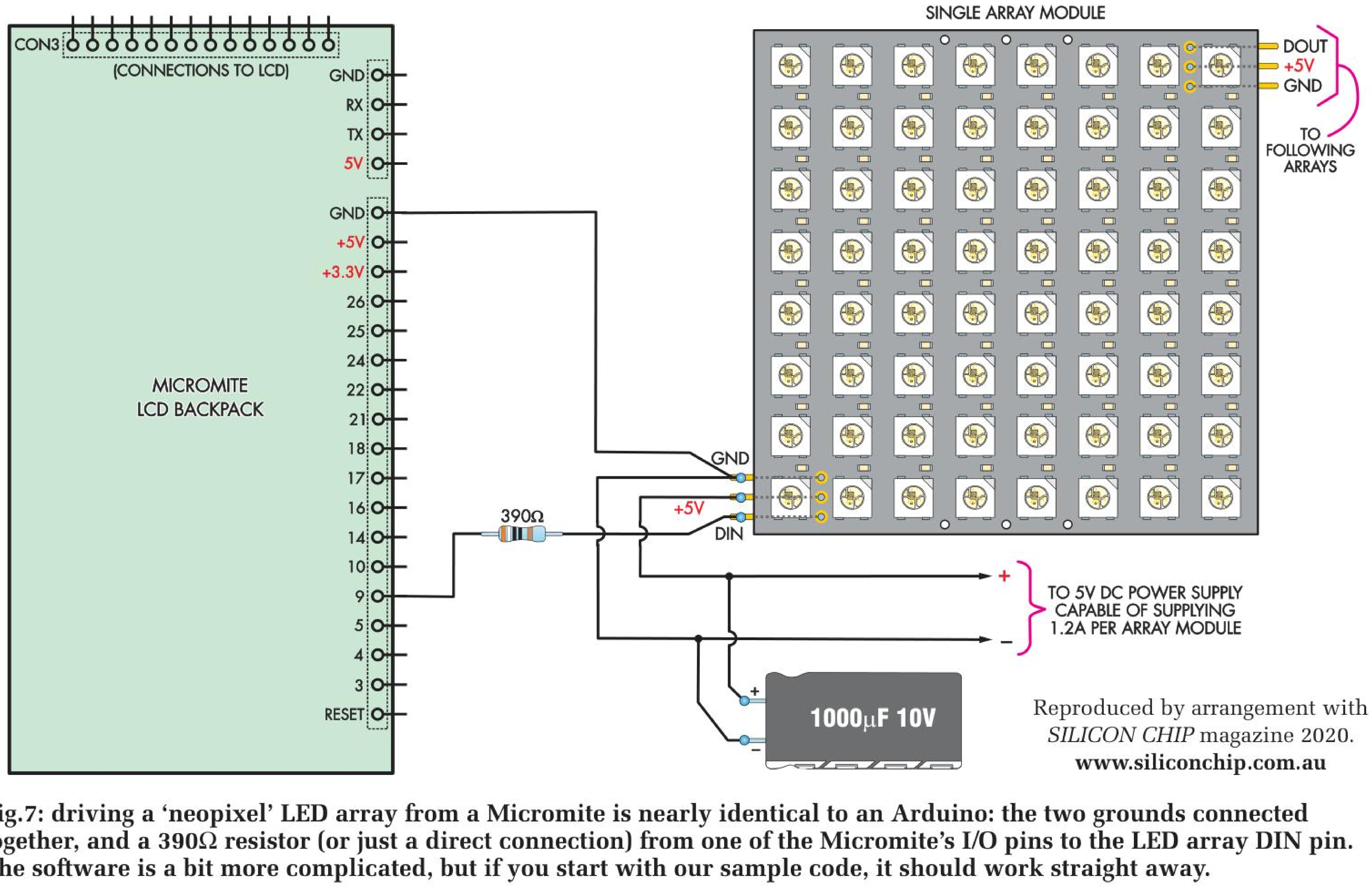
I then proceeded to write an expanded version of my original MMBasic test program to provide readers with a suitable demo program to run on a Micromite. This program is called **RGB LED matrix test program.bas**, and again, you can download it from: the January 2021 page of the *PE* website.

This program displays a 'rainbow' of coloured stripes on the 64-LED SW2812B/SK6812 module. It then clears the module's display for another five seconds before repeating the cycle. While simple, again I hope it will give you a good idea as to how a Micromite can be used to drive these LED modules.

To achieve different kinds of display (including dynamic displays), all you need to do is use the MMBasic part of the program to change the 'pixel' data stored in the `colours()` array.

You can find some useful information on this module in the following links:

http://bit.ly/pe-jan21-8x8a
http://bit.ly/pe-jan21-8x8b
http://bit.ly/pe-jan21-8x8c


SINGLE ARRAY MODULE
DOUT
+5V
GND
TO FOLLOWING ARRAYS
CON3
(CONNECTIONS TO LCD)
GND
RX
TX
5V
GND
+5V
+3.3V
26
25
24
22
21
18
17
16
14
10
9
5
4
3
RESET
MICROMITE LCD BACKPACK
390Ω
GND
+5V
DIN
+
–
TO 5V DC POWER SUPPLY CAPABLE OF SUPPLYING 1.2A PER ARRAY MODULE
1000µF 10V

Reproduced by arrangement with
SILICON CHIP magazine 2020.
www.siliconchip.com.au

Fig.7: driving a 'neopixel' LED array from a Micromite is nearly identical to an Arduino: the two grounds connected together, and a 390Ω resistor (or just a direct connection) from one of the Micromite's I/O pins to the LED array DIN pin. The software is a bit more complicated, but if you start with our sample code, it should work straight away.

KickStart

by Mike Tooley

Part 1: MOSFET switching devices in linear applications – introducing the 2N7000 ‘Swiss Army Knife’

Our occasional *KickStart* series aims to show readers how to use readily available low-cost components and devices to solve a wide range of common problems in the shortest possible time. Each of the examples and projects can be completed in no more than a couple of hours using off-the-shelf parts. As well as briefly explaining the underlying principles and technology used, the series will provide you with a variety of representative solutions and examples, along with just enough information to be able to adapt and extend them for their own use.

This first part shows you how to use a low-cost MOSFET switching device in a variety of linear applications. In keeping with the *KickStart* philosophy, we’ve provided sufficient information for you to be able to design and build your own circuits using this handy semiconductor device.

MOSFET devices are available in various forms, including N-type and P-type, and enhancement or depletion mode types. The 2N7000 is an N-type MOSFET designed for enhancement mode operation (see Fig.1.1). This simply means that to turn the device ‘on’ (ie, to make it conduct) it is necessary to apply a positive voltage between the gate and source of the device. For the 2N7000, the required gate-source voltage is in the range of 2V to 3V. The device will conduct very heavily when the gate-source voltage exceeds about 3V, in which case the resistance between drain and source (R_{DS}) will fall to the very low value required for switching applications. However, with gate-source voltages (V_{GS}) of less than 2.5V the device can be used in a wide variety of *linear* applications. This is where this incredibly versatile ‘Swiss Army Knife’ finds a whole new variety of applications!

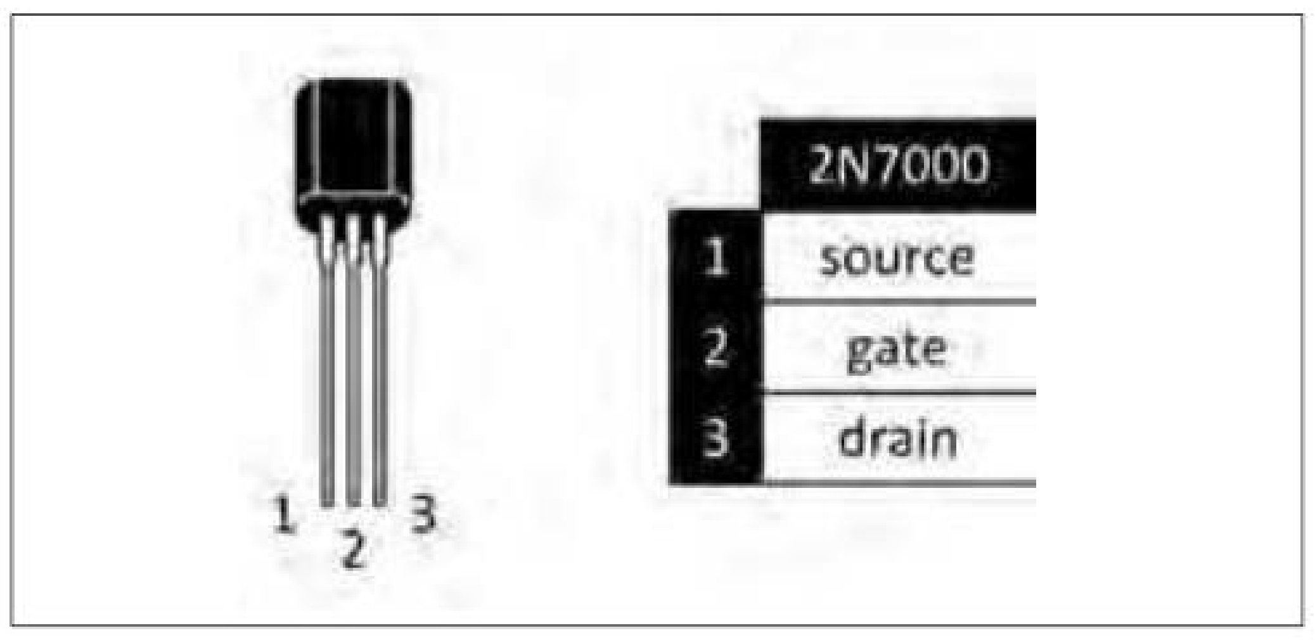


Fig.1.2. 2N7000 pin connections.

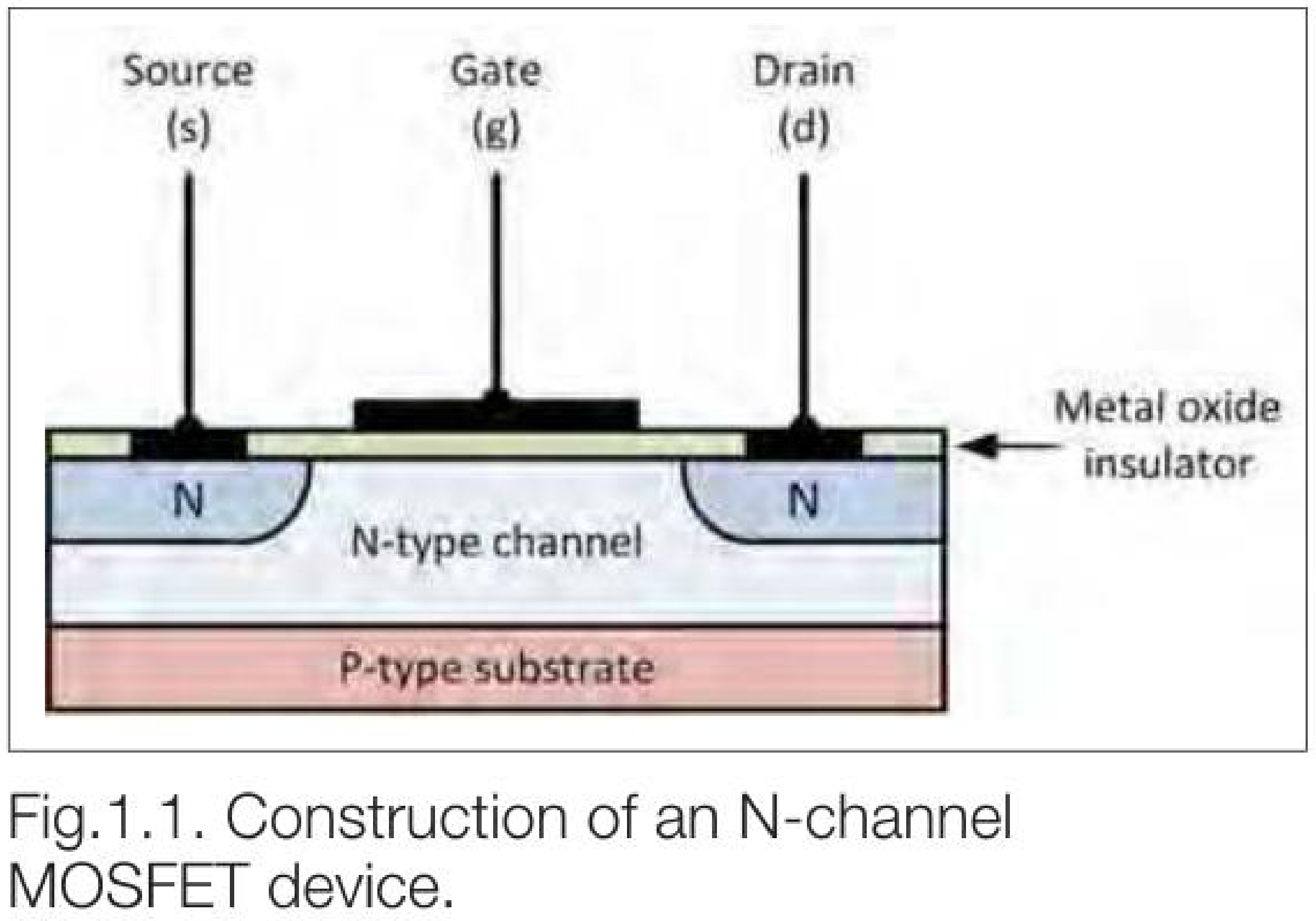


Fig.1.1. Construction of an N-channel MOSFET device.

MOSFET basics

The construction of a typical N-channel MOSFET device is shown in Fig.1.1. The device consists of a series of semiconductor layers onto which an insulating metal-oxide layer is deposited. Conduction takes place between source and drain in a narrow N-type channel. The degree of conduction in this region is dependent on the positive potential present on the gate terminal. As the gate-source voltage (V_{GS}) is raised beyond the threshold for conduction (usually above about 2V for the 2N7000) conduction increases and the drain and source currents (which are identical) increase as a result. Thus, the voltage present between the gate and source controls the current flowing in the drain.

In manufacturers’ data sheets, device properties are usually summarised both in the form of tabulated data and as a series of characteristic curves. Key data for the 2N7000 is shown in Table 1.1, and the two most important characteristic curves are shown Fig.1.4 and Fig.1.5. The test circuit that we used to obtain these plots is shown in Fig.1.3. Note that the drain current (I_D) needs to be kept to a safe value in order to limit the total power dissipation of 350mW quoted in Table 1.1.

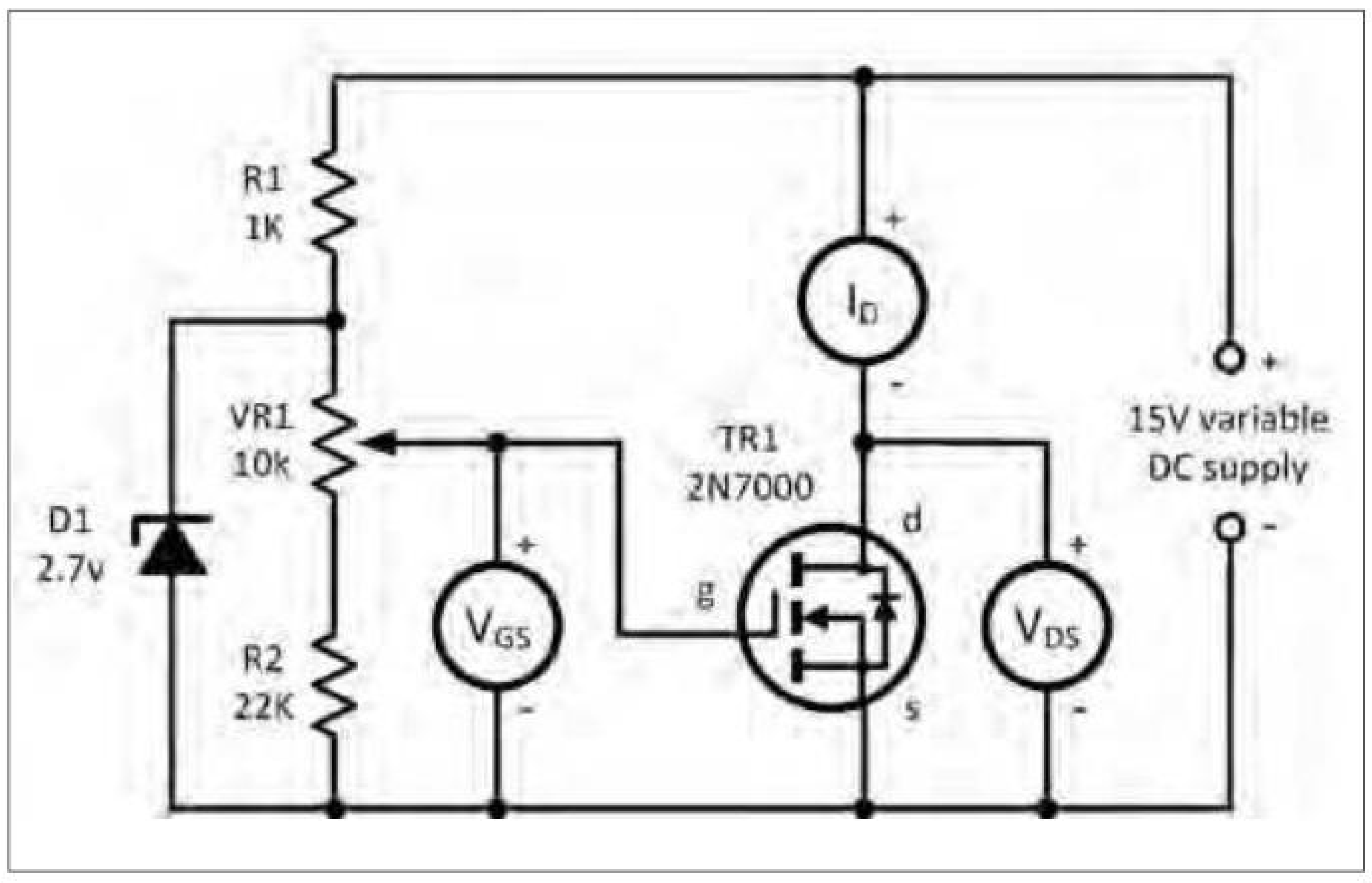


Fig.1.3. N-channel MOSFET test circuit.

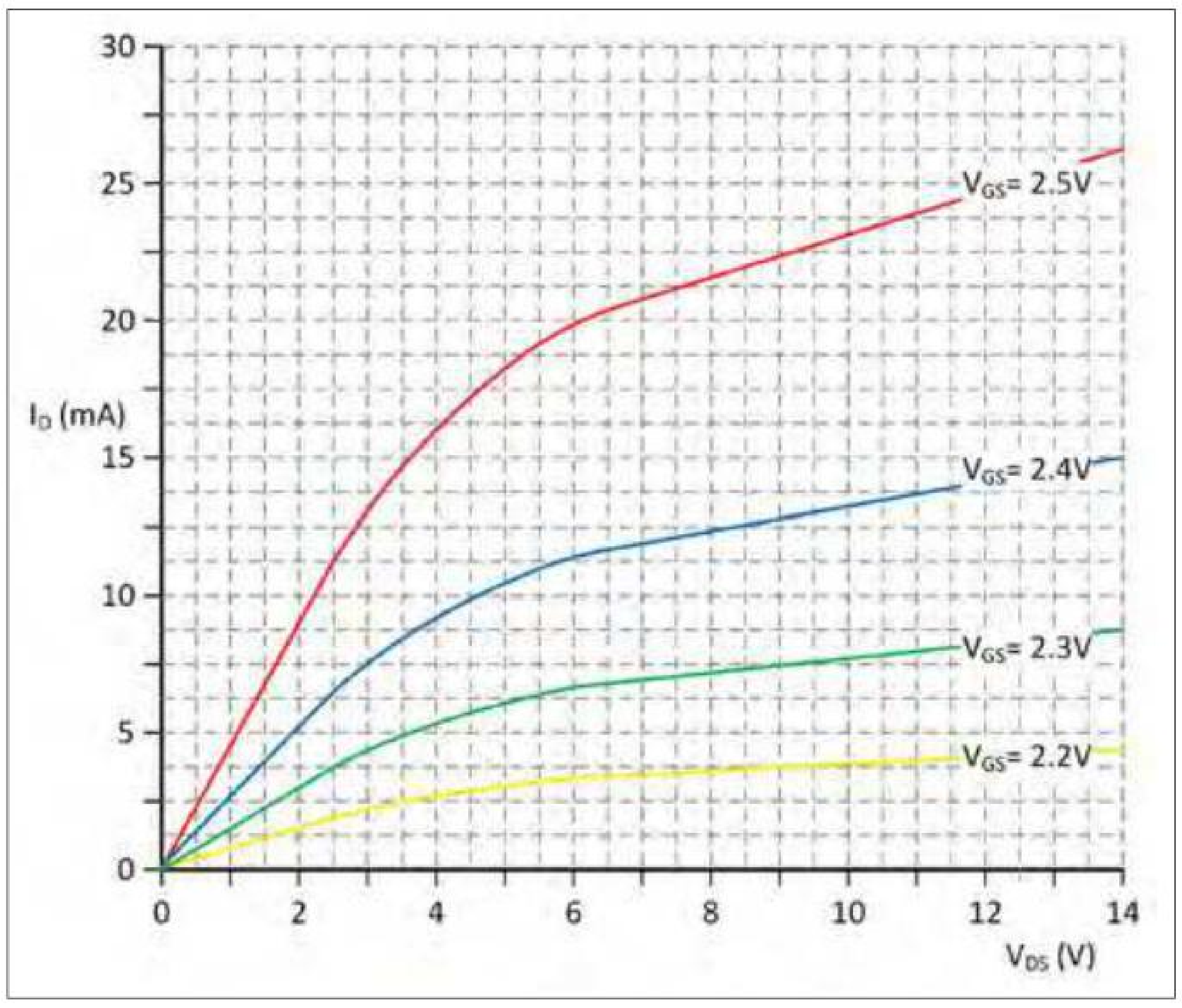


Fig.1.4. Output (drain) characteristics for a 2N7000 operating under small-signal conditions.

The output characteristics (shown in Fig.1.4) consist of a series of plots showing the variation of drain current (I_D) with drain-source voltage (V_{DS}) for different values of gate-source voltage (V_{GS}). There are a few things to note about these curves. First, we have plotted these curves using relatively small values of drain current that would be used in linear (rather than switching) applications. Second, note the bend that occurs for values of V_{DS} below about 2V. .

Table 1.1: Key data for the 2N7000

Specification	Abbrev.	Value
Maximum drain-source voltage	$V_{DS(max)}$	60V
Maximum drain-gate voltage	$V_{DG(max)}$	60V
Maximum gate-source voltage	$V_{GS(max)}$	±20V
Maximum drain current	$I_{D(max)}$	200mA
Maximum total power dissipation	$P_{D(max)}$	350mW
Maximum gate-source threshold voltage	$V_{GS(th.max)}$	0.8V
Minimum gate-source threshold voltage	$V_{GS(th.min)}$	3V
Minimum forward transconductance	g_{fs}	0.1mS
Maximum input capacitance	C_{iss}	60pF

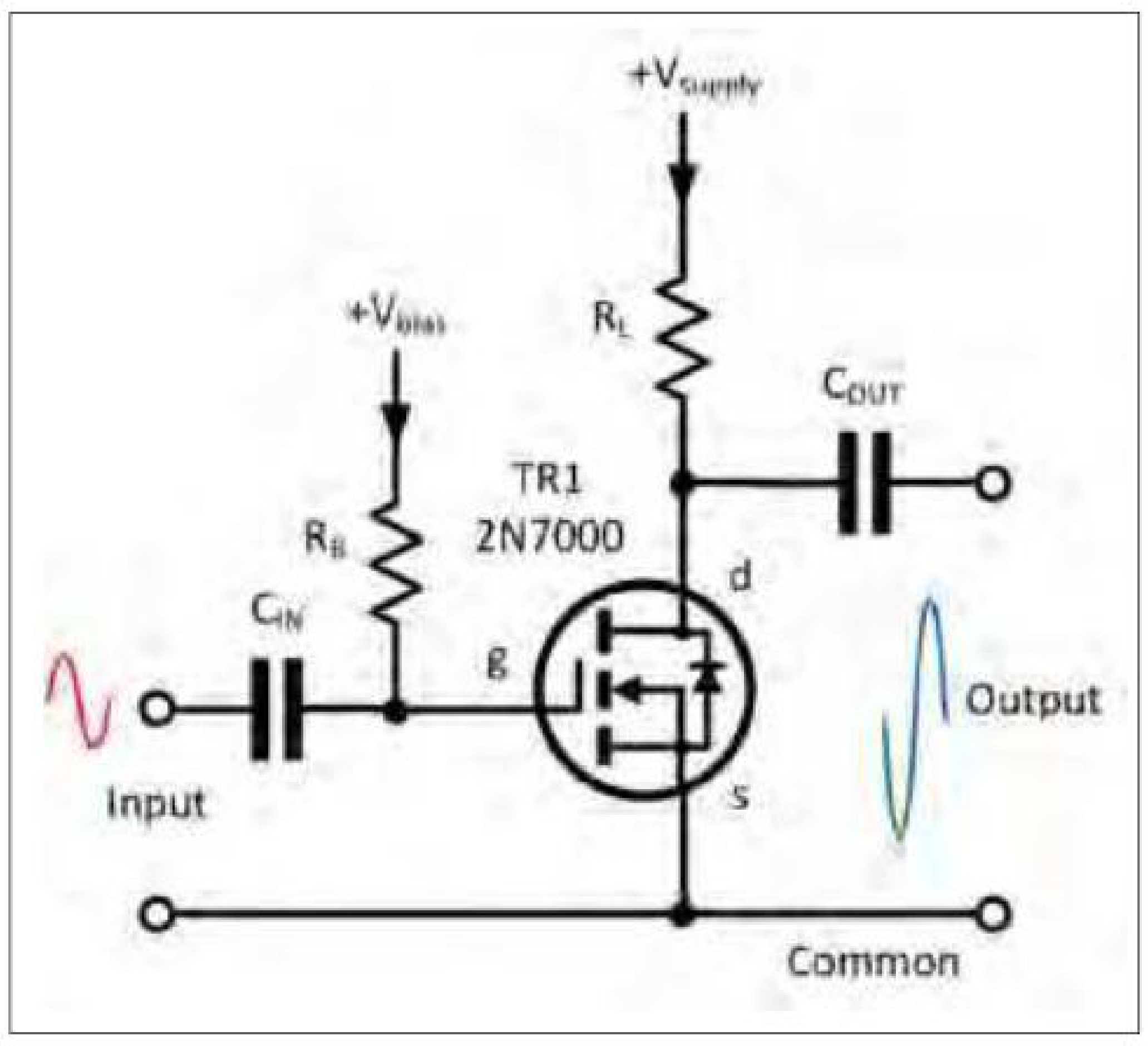


Fig.1.6. Bias and load arrangements for a simple common-source MOSFET amplifier stage.

The transfer characteristics (shown in Fig.1.5) provide us with another family of plots. They show how the drain current (I_D) varies with gate-source voltage (V_{GS}) for different values of drain-source voltage (V_{DS}). The three curves shown correspond to V_{DS} values of 3V, 6V and 11V. As can be seen, they are all very similar. It is worth noting that we have plotted these curves using relatively small values of drain current (I_D) and, as with the output characteristic curves shown in Fig.1.4, the values of drain current are very much less than those shown in the data sheets published by semiconductor manufacturers.

The slope of the transfer characteristic is of particular significance. This is known as the forward transfer conductance (g_{fs}) and is defined as a small change in drain current (ΔI_D) divided by the corresponding small change in gate-source voltage, (ΔV_{GS}) at a given value of drain-source voltage (V_{DS}), thus:

$$g_{fs} = \frac{\Delta I_D}{\Delta V_{GS}}$$

The value of g_{fs} is quoted using the unit of conductance (siemen (S), or even 'mho' (which is simply 'ohm' backwards!) by some manufacturers). However, since the values are usually fairly small, we use mS (milli-siemen) or mmho instead.

Putting this into the context of Fig.1.4, a change in drain current from 6mA to 10.9mA will be produced by a change in gate-source voltage from 2.3V to 2.4V. Hence, at this point on the transfer characteristic when V_{DS} = 6V, we can determine the forward transfer conductance from:

$$g_{fs} = \frac{(11-6.0)\,\text{mA}}{(2.4-2.3)\,\text{V}} = \frac{5\text{mA}}{0.1\text{V}} = 50\ \text{mS}$$

Using MOSFET devices in linear applications

As previously mentioned, Fig.1.6 shows the basic components that are used in a simple common-source MOSFET amplifier stage.

In order to use a 2N7000 in linear mode it is necessary to provide a gate-source bias voltage of around 2V. The gate bias (V_{bias}) can be applied to the gate via a high-value resistor, R_B. The value of R_B is not critical but is usually in the range 100kΩ to 1MΩ. The output voltage is developed across a load resistor (R_L) of suitable value connected in the drain circuit. Capacitors C_{IN} and C_{OUT} (respectively) are used to couple the AC signal into and out of the stage.

Load lines

To understand how the load resistor works it is worth taking a look at Fig.1.7 which shows how a load line can be superimposed on the output characteristics that we met earlier. This particular load line corresponds to a value of 500Ω for R_L. The two ends of the load line correspond to the extreme values that would occur when TR1 is either fully conducting (on) or non-conducting (off). The slope of the load line is thus the inverse of R_L. The operating point is the point at which corresponding values of V_{DS} and I_D will occur when no signal is applied (this is sometimes referred to as the 'quiescent condition'). When a signal is applied to the stage, V_{GS} will change and the

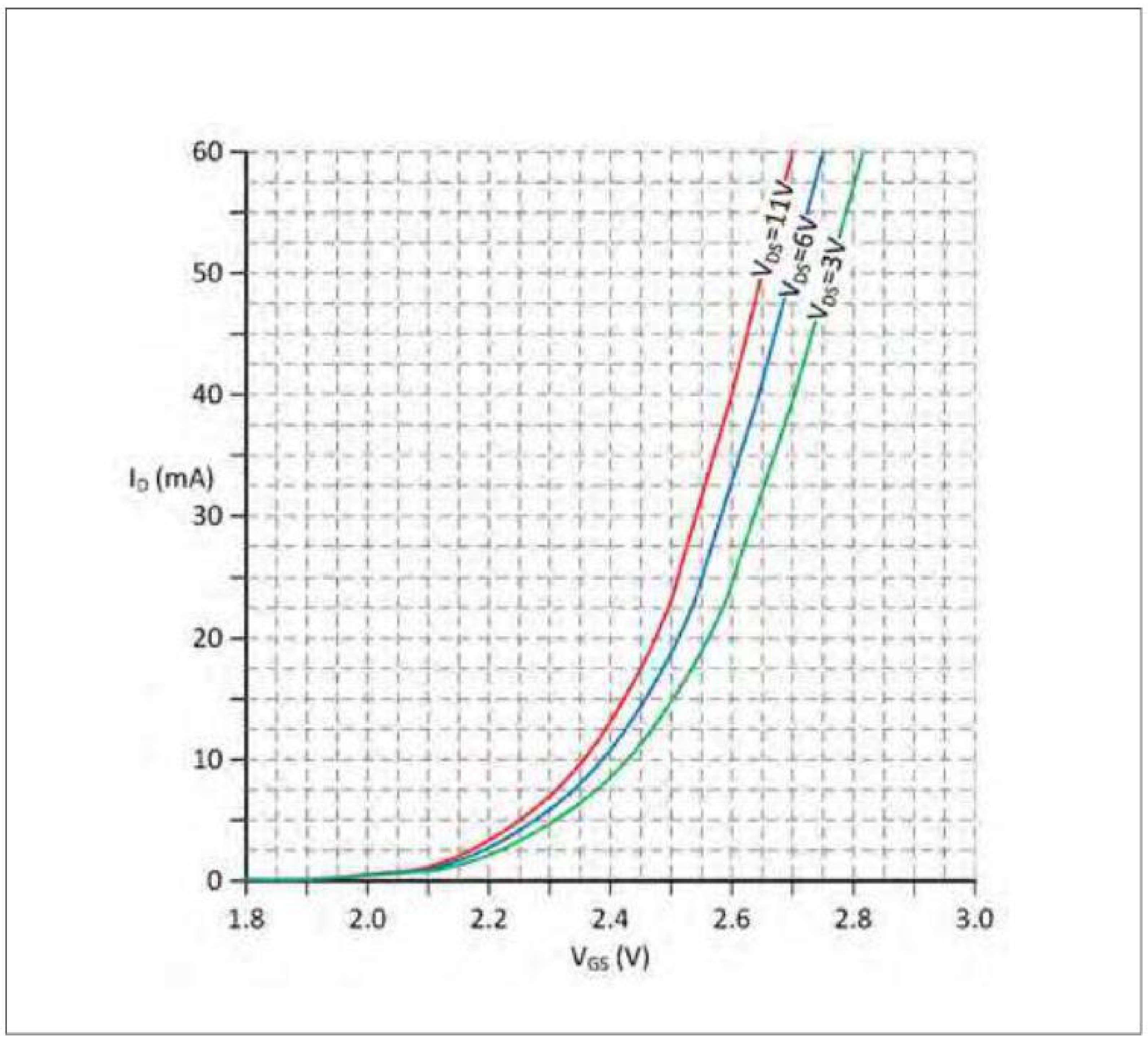


Fig.1.5 Transfer characteristics for a 2N7000 operating under small-signal conditions.

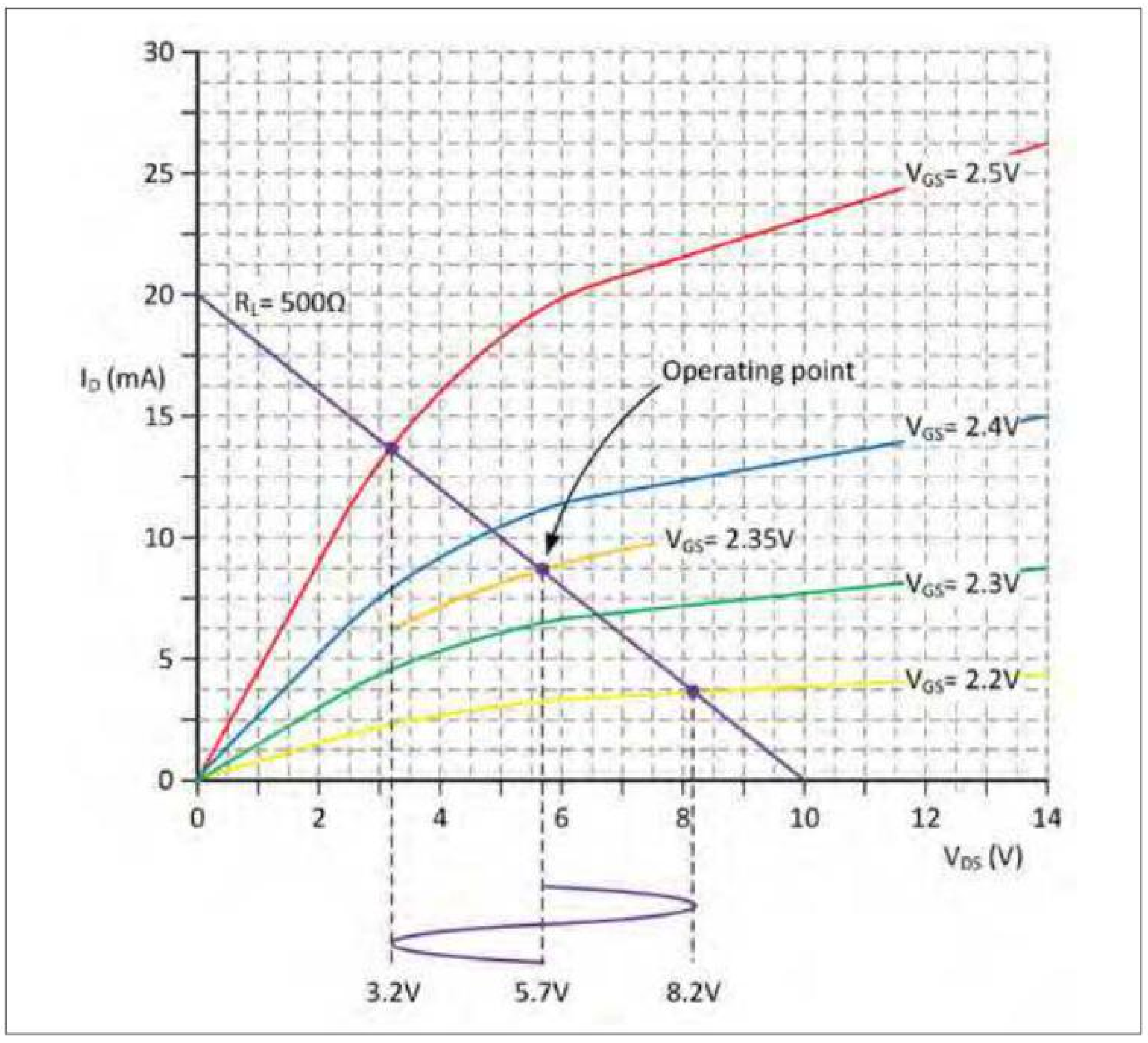


Fig.1.7. Load line superimposed on the 2N7000 output characteristics.

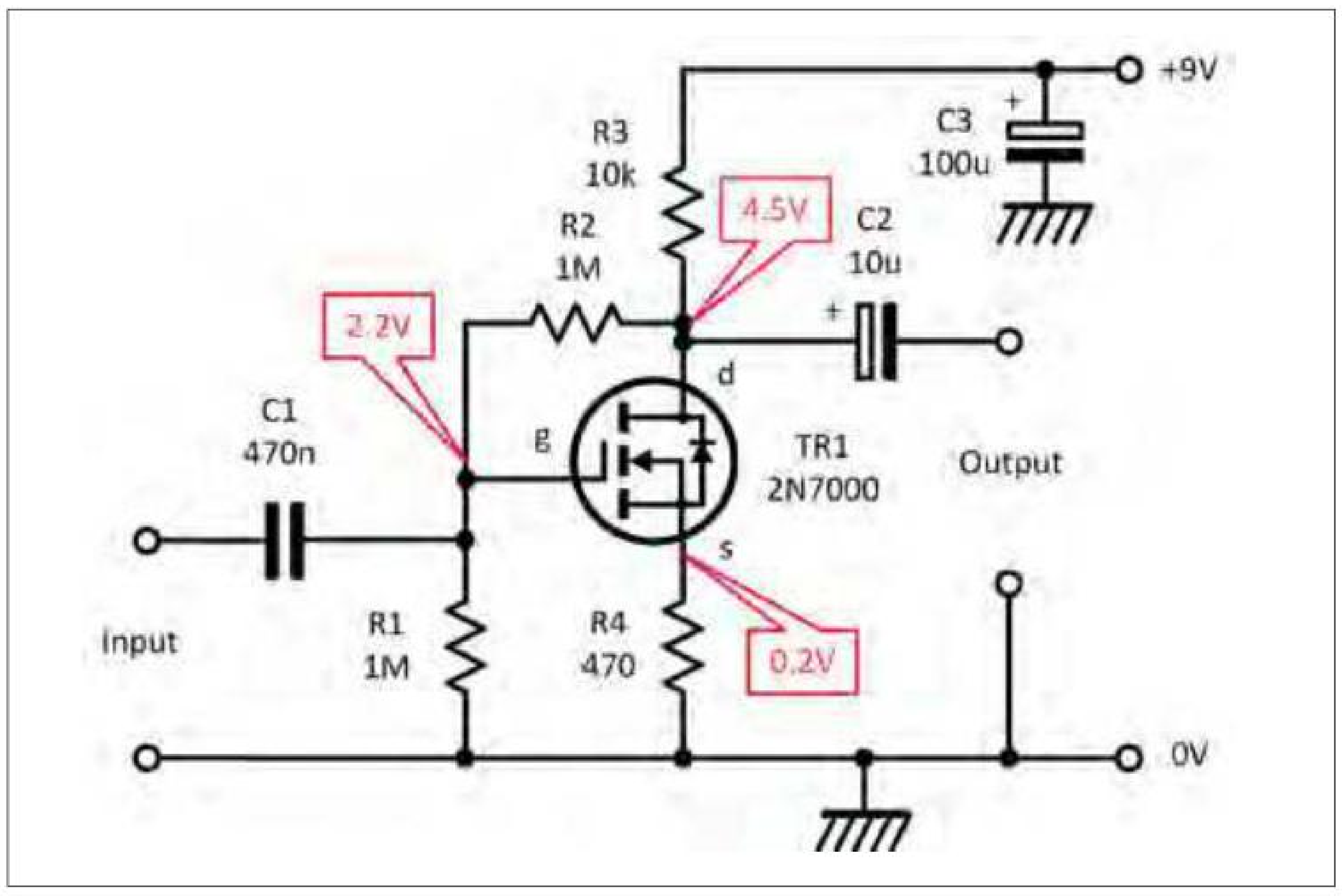

+9V
R3
10k
R2
1M
4.5V
C3
100u
C2
10u
2.2V
C1
470n
g
d
s
TR1
2N7000
Output
Input
R1
1M
R4
470
0.2V
0V

Fig.1.8. A basic amplifier using the 2N7000.

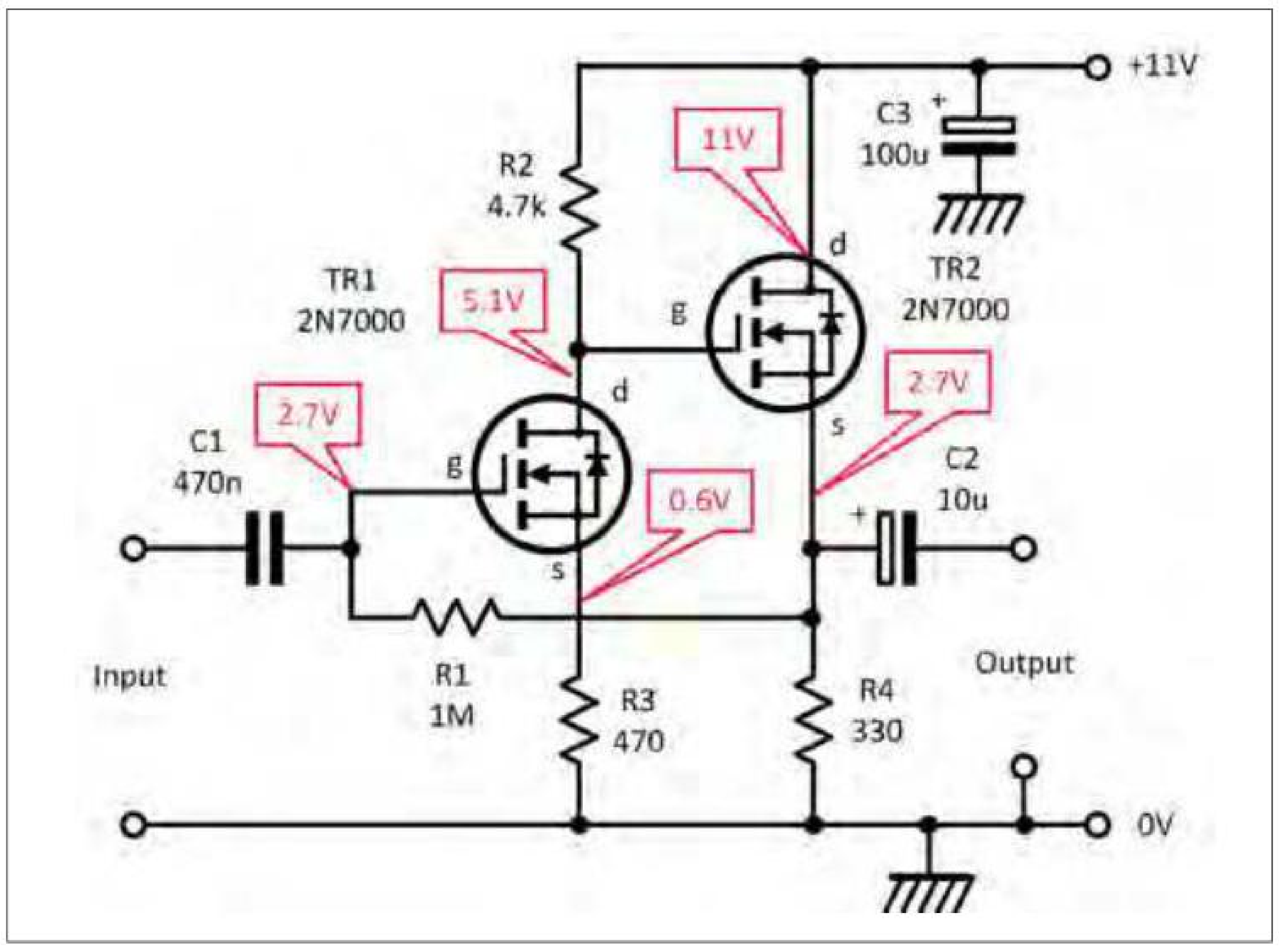

+11V
R2
4.7k
11V
C3
100u
TR1
2N7000
5.1V
g
d
TR2
2N7000
2.7V
C1
470n
g
d
s
0.6V
s
2.7V
C2
10u
Input
R1
1M
R3
470
R4
330
Output
0V

Fig.1.10. An improved 'gain block'.

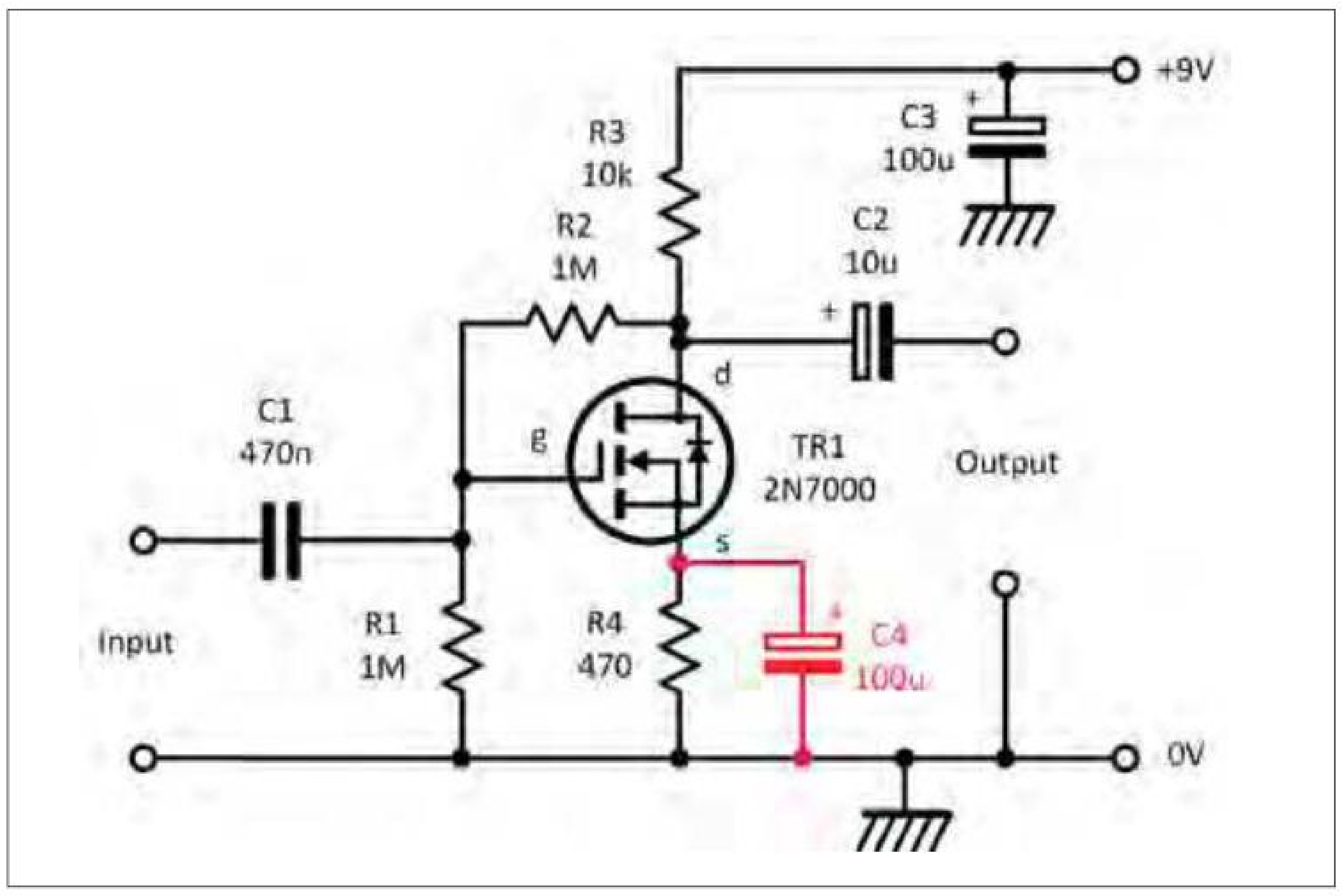

+9V
R3
10k
C3
100u
R2
1M
C2
10u
C1
470n
g
d
s
TR1
2N7000
Output
Input
R1
1M
R4
470
C4
100u
0V

Fig.1.9. An amplifier stage with a gain of about 40.

operating point will move up and down the load line. It is then possible to read off corresponding values of V_{DS} and use them to infer the shape of the output voltage waveform, as shown in Fig.1.7.

With the conditions indicated on Fig.1.7, if a 300mV_{pk-pk} signal is superimposed on a standing bias voltage of 2.35V, the value of drain-source voltage (V_{DS}) will swing down to a minimum of 3.2V and up to a maximum of 8.2V. Hence an input of 350mV_{pk-pk} will produce an output of 5V_{pk-pk} corresponding to a modest voltage gain (A_V) of a little over 14. The required bias voltage (2.35V in the case of Fig.1.7) can be obtained in various ways, but one of the most basic arrangements is nothing more than a resistive potential divider derived either from the DC supply rail or from the drain connection.

A basic 2N7000 amplifier

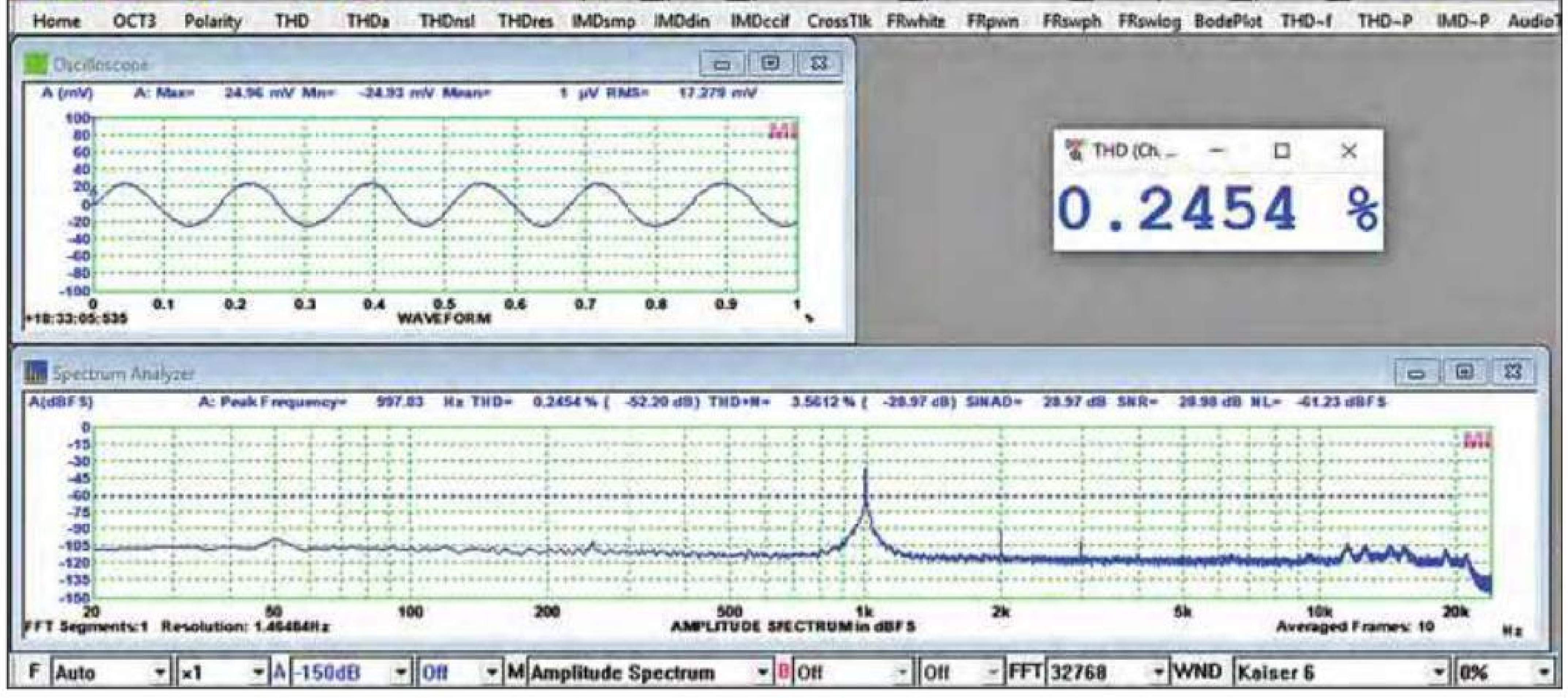
A basic common-source amplifier is shown in Fig.1.8. The load for the stage is provided by R3, while the gate-source bias voltage (which needs to be approximately 2.2V for optimum operating conditions in this circuit) is defined by the potential divider formed by R1 and R2. Optimum operating conditions are those that will give the maximum undistorted output voltage swing. This condition occurs when the drain voltage is about half the supply voltage (ie, around 4.5V neglecting the small voltage dropped across R4). Typical operating voltages are shown on Fig.1.8, but in practice, and due to variations in the characteristics of individual components and MOSFET devices, variations of around ±10% can be expected. If significant differences are encountered, then the values of R1 and/or R2 can be changed accordingly.

The voltage gain of the stage is approximately 14. This means that a signal input of 100mV_{pk-pk} will result in an output of 1.4V_{pk-pk}. If more gain is required then a capacitor can be introduced in parallel with R4, as shown in Fig.1.9. This capacitor will bypass the signal voltage component that appears across R4, reducing the amount of negative feedback and increasing the voltage gain as a result. With R4 bypassed to signals, the circuit shown in Fig.1.9 provides a voltage gain of around 40 so that an input of 100mV_{pk-pk} will produce an output of 4V_{pk-pk}, and so on. The input resistance of both circuits is approximately 500kΩ (determined largely by R1 and R2) and the measured frequency response extends from less than 10Hz to well over 200kHz at the –3dB points.

Negative feedback and bias stabilisation

The circuit of Fig.1.8 employs two forms of negative feedback: shunt voltage feedback from drain to gate via R3 and series current feedback using R4 in the connection from source to ground. These two feedback loops help to stabilise the DC operating conditions, ensuring that the operating conditions remain within the desired range despite variations in MOSFET parameters and temperature. They also make a significant improvement in the overall linearity of amplifier.

Operation of the shunt feedback loop is as follows. If the gate-source voltage increases the drain current will increase as a direct consequence. The increase in drain current will result in an increased potential drop across the load resistance


Multi-Instrument Pro 3.9 - <Microphone (USB PnP Sound Devic>
File
Setting
Instrument
Window
Help
Trigger
Normal
A
Up
0%
0%
NIL
Sample
48kHz
A
16Bit
Point
48000
Roll
Record
Auto
AC
AC
±1V
±1V
Probe
x1
x1
Home
OCT3
Polarity
THD
THDa
THDnsl
THDres
IMDsmp
IMDdin
IMDccif
CrossTlk
FRwhite
FRpwn
FRswph
FRswlog
BodePlot
THD-f
THD-P
IMD-P
Oscilloscope
A (mV)
100
80
60
40
20
0
-20
-40
-60
-80
-100
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
s
WAVEFORM
+18:33:05:535
THD (Ch...
Spectrum Analyzer
A(dBFS)
0
-15
-30
-45
-60
-75
-90
-105
-120
-135
-150
20
50
100
200
500
1k
2k
5k
10k
20k
Hz
AMPLITUDE SPECTRUM in dBFS
F
Auto
x1
A
-150dB
Off
M
Amplitude Spectrum
B
Off
Off
FFT
32768
WND
Kaiser 6
0%

A: Max= 24.96 mV Min= -24.93 mV Mean= 1 µV RMS= 17.279 mV

0.2454 %

A: Peak Frequency= 997.83 Hz THD= 0.2454 % (-52.20 dB) THD+N= 3.5612 % (-28.97 dB) SINAD= 28.97 dB SNR= 28.98 dB NL= -61.23 dBFS

FFT Segments:1 Resolution: 1.46484Hz

Averaged Frames: 10

Fig.1.11. Using Virtins Multi-Instrument PC-based software to analyse and measure the total harmonic distortion (THD) produced by the circuit of Fig.1.10.

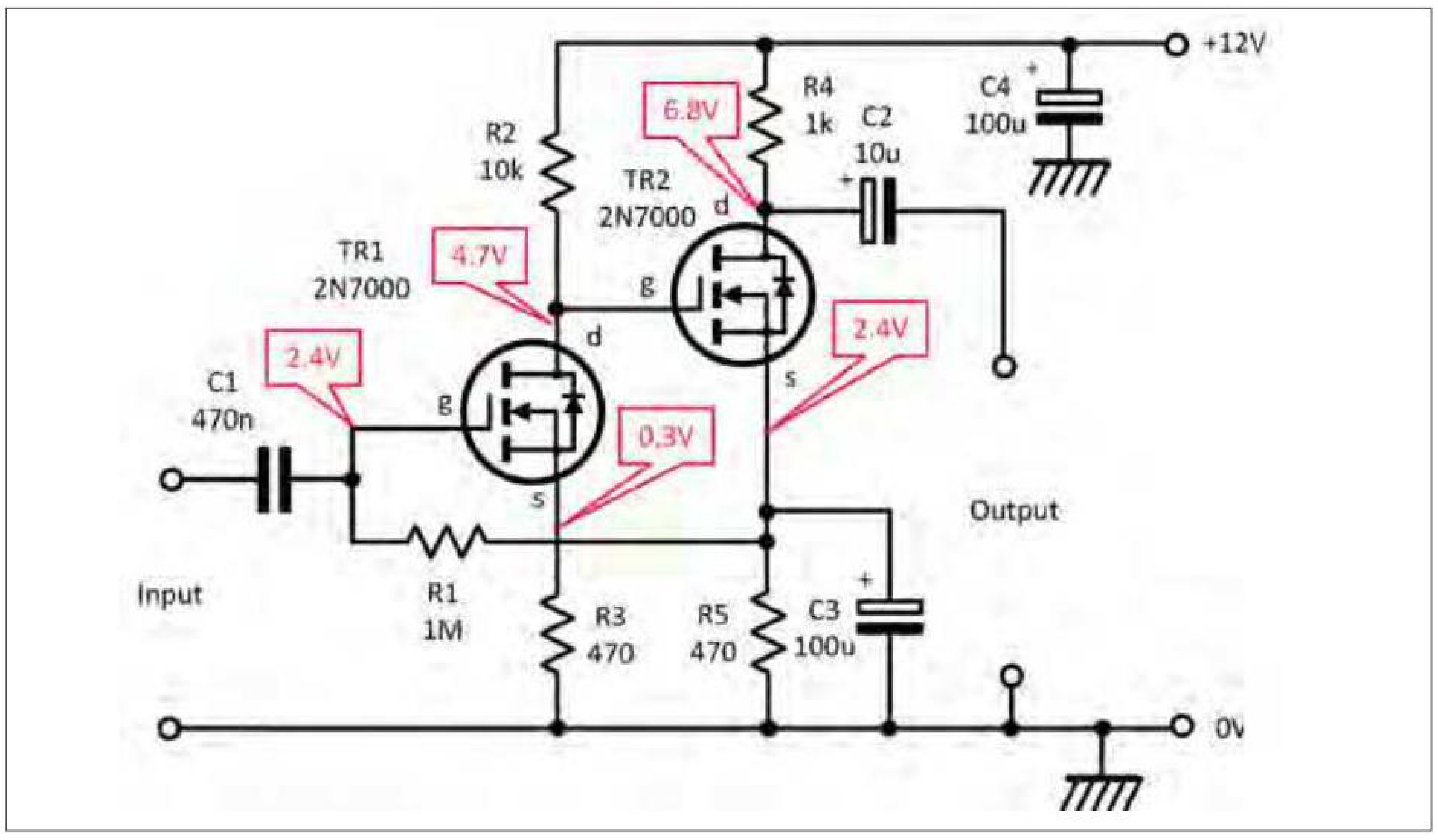

+12V
R4
1k
C2
10u
C4
100u
R2
10k
6.8V
TR2
2N7000
TR1
2N7000
4.7V
2.4V
2.4V
C1
470n
0.3V
g
d
s
Output
Input
R1
1M
R3
470
R5
470
C3
100u
0V

Fig.1.12. Using Virtins Multi-Instrument PC-based software to analyse and measure the total harmonic distortion (THD) produced by the circuit of Fig.1.10.

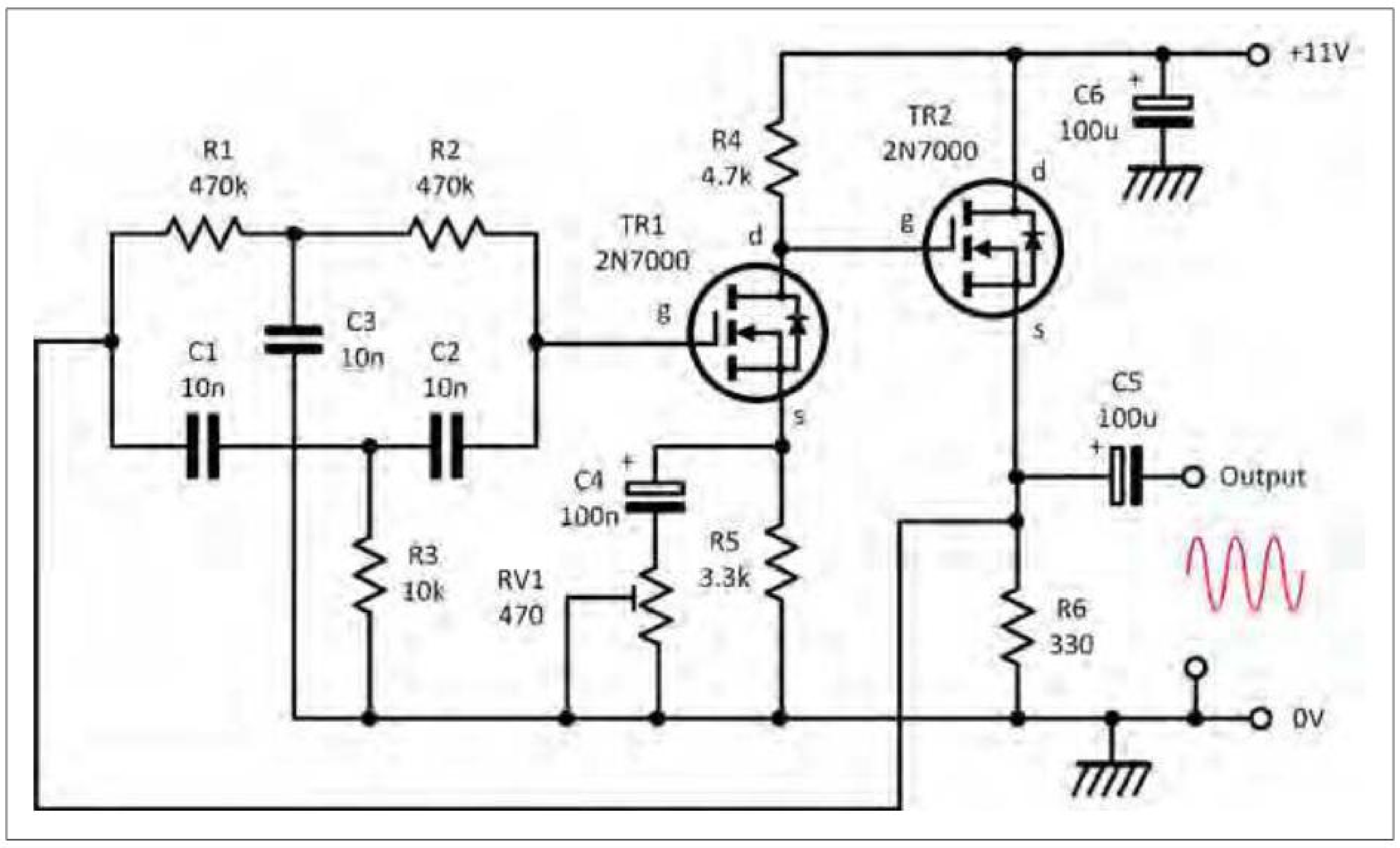

+11V
R1
470k
R2
470k
R4
4.7k
TR2
2N7000
C6
100u
TR1
2N7000
C3
10n
C1
10n
C2
10n
C5
100u
Output
C4
100n
R5
3.3k
R3
10k
RV1
470
R6
330
0V

Fig.1.13. A twin-tee oscillator based on the two-stage gain block in Fig.1.10.

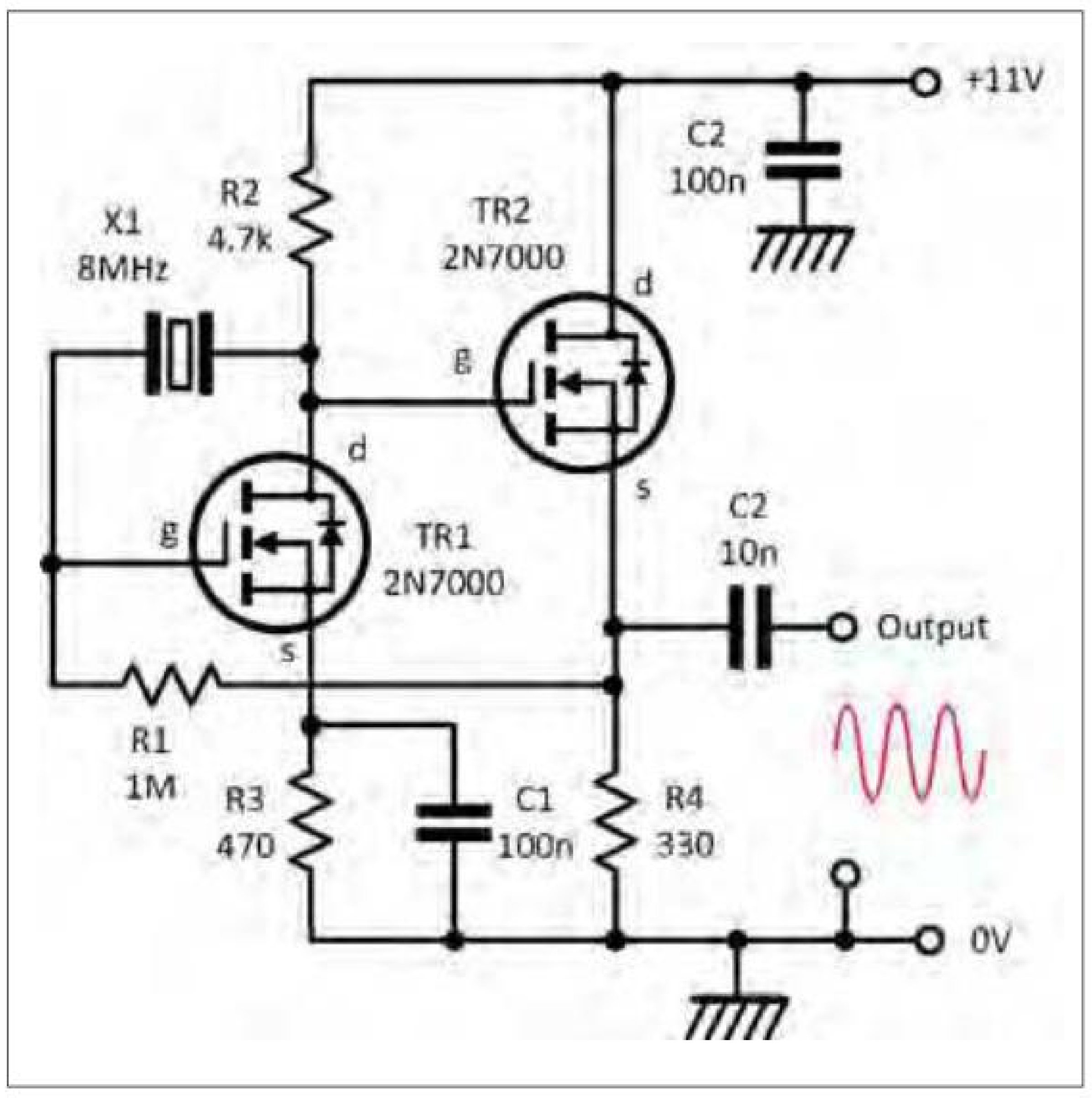

+11V
X1
8MHz
R2
4.7k
TR2
2N7000
C2
100n
TR1
2N7000
C2
10n
Output
R1
1M
R3
470
C1
100n
R4
330
0V

Fig.1.15. 8MHz crystal oscillator using a 2N7000.

(R_L). As a result, the drain voltage will fall, which, in turn, reduces the gate bias voltage. Capacitors, C1 and C2 respectively, are used to couple signals into and out of the stage. Put simply, these capacitors pass AC signals but keep the DC gate bias and drain voltages safely within the amplifier stage.

An improved 'gain block'

An improved two-stage amplifier is shown in Fig.1.10. The first stage acts as a common-source amplifier with a high input impedance; the second as a source-follower with a low output impedance. The two-stage 'gain block' arrangement reduces the effect of output loading on the drain load of the first stage (TR1) and produces a nominal voltage gain of 10. The measured frequency response extends from around 4.5Hz to 450kHz. Notice how negative feedback (via R1) is applied over both stages (ie, from the source of the second stage to the gate of the first stage).

Distortion

No amplifier is perfectly linear, and no amplifier can provide a perfect representation of its input. The result of non-linearity is distortion and, while a small level of distortion may be undetectable by ear, designers usually go to great lengths to reduce the level of distortion introduced by an amplifier.

It is important to note that, as an amplifier is driven harder, the distortion that it produces will increase. The circuit shown in Fig.1.10 typically produces around 0.25% total harmonic distortion (THD) when a signal of $100mV_{pk\text{-}pk}$ is applied, as shown in Fig.1.11. (Note that the THD increases to around 6% for the circuit of Fig.1.9 with the same signal.) It is therefore important to avoid over-driving an amplifier, particularly where the internal gain is appreciable.

A high-gain amplifier

A high-gain amplifier is shown in Fig.1.12. This arrangement uses two cascaded common-source amplifier stages. This circuit produces a typical voltage gain of 250 with C3 not fitted, increasing to around 370 with C3 fitted (as shown in Fig.1.12). Note that signal levels must be kept low to avoid over-driving the second stage.

Sinusoidal oscillators

As well as its use in small-signal amplifiers, the 2N7000 can be used in a wide variety of sinusoidal oscillator applications. Fig.1.13 shows a simple twin-tee oscillator based on the two-stage gain block in Fig.1.10. The circuit provides an output of around $8V_{pk\text{-}pk}$ at approximately 100Hz. The frequency of

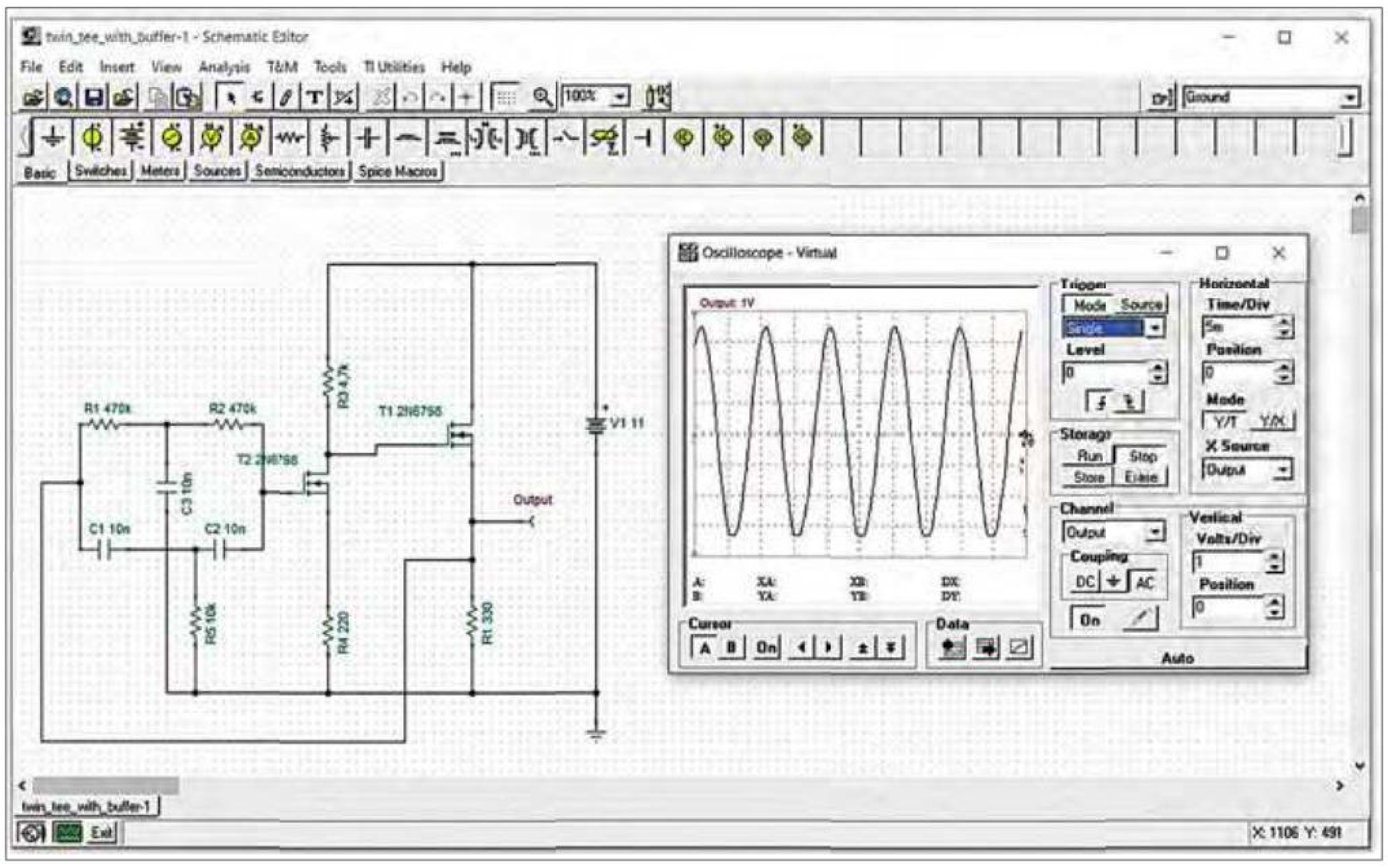

twin_tee_with_buffer-1 - Schematic Editor
File
Edit
Insert
View
Analysis
T&M
Tools
TI Utilities
Help
100%
Ground
Basic
Switches
Meters
Sources
Semiconductors
Spice Macros
R1 470k
R2 470k
R3 4.7k
T1 2N6798
T2 2N6798
V1 11
C3 10n
C1 10n
C2 10n
Output
R5 10k
R4 220
R1 330
Oscilloscope - Virtual
Output: 1V
Trigger
Mode
Source
Single
Level
Horizontal
Time/Div
Position
Mode
Y/T
Y/X
X Source
Output
Storage
Run
Stop
Store
Erase
Channel
Output
Coupling
DC
AC
Vertical
Volts/Div
Position
Cursor
Data
On
Auto
twin_tee_with_buffer-1
Exit
X: 1106 Y: 491

Fig.1.14. Using Tina Pro to check and optimise component values for the twin-tee oscillator shown in Fig.1.13.

Table 1.2: Going Further with the 2N7000 MOSFET

Topic	Source	Notes
2N7000 MOSFET	The 2N7000 is available from many suppliers, including CPC/Farnell, Mouser, RS Components and numerous online suppliers. Data sheets can be downloaded from manufacturers' websites including ON Semiconductor, Fairchild and Vishay	Prices range from a few pence to about 50p depending on the quantity purchased
Audio amplifiers	For all your audio amplifier requirements, the *PE* column *Audio Out* provides a treasure chest of tips, hints, designs and ideas, drawing on Jake Rothman's decades of experience in teaching, designing and building a huge range of audio circuits.	All *Audio Out* columns are available via back issues at: **www.electronpublishing.com**
Circuit simulation	The author's own book, *Electronic Circuits: Fundamentals and Applications* (Fifth Edition 2020 published by Routledge 9780367421984) provides an introduction to circuit simulation based on the popular SPICE-based Tina Pro software	Tina software can be obtained from www.tina.com. A cut-down version can be freely downloaded from Texas Instruments at: www.ti.com/tool/TINA-TI
Distortion	A detailed explanation of different types of distortion can be found in Part 8 of Electronics Teach-In 7	**www.electronpublishing.com/product/electronics-teach-in-7**
Negative feedback	For a useful introduction and relevant theory see Part 8 of Electronics Teach-In 7	**www.electronpublishing.com/product/electronics-teach-in-7**
Transistor characteristics and load lines	The author's own book, *Electronic Circuits: Fundamentals and Applications* (see above) provides an introduction to transistor characteristics, load lines and amplifiers.	

operation can be altered by varying the values used in the twin-tee network (see Going further below). Note that popular Spice-based simulation software can often be used to model and optimise the operation of MOSFET circuits, as shown in Fig.1.14.

Finally, Fig.1.15 shows a crystal-controlled oscillator based on a single 2N7000 device. This provides a typical output of $2V_{pk\text{-}pk}$ and has been tested with quartz crystals between 2MHz and 20MHz. This arrangement forms the basis of the author's own simple Crystal Checker circuit, which he uses in his workshop.

Going further

Our *Going further* table (opposite) will help you locate the component parts and further information that will allow you to quickly progress with your own designs and modifications. It also provides you with background reading that will help you get up to speed with the necessary underpinning knowledge for key topics discussed.

GET THE LATEST COPY OF OUR TEACH-IN SERIES
AVAILABLE NOW!

Order direct from Electron Publishing

PRICE £8.99
(includes P&P to UK if ordered direct from us)

ELECTRONICS TEACH-IN 9
£8.99
FREE CD-ROM
GET TESTING!
PIC n' Mix
Teach-In 9
Get Testing

Teach-In 9 – Get Testing!

This series of articles provides a broad-based introduction to choosing and using a wide range of test gear, how to get the best out of each item and the pitfalls to avoid. It provides hints and tips on using, and – just as importantly – interpreting the results that you get. The series deals with familiar test gear as well as equipment designed for more specialised applications.

The articles have been designed to have the broadest possible appeal and are applicable to all branches of electronics. The series crosses the boundaries of analogue and digital electronics with applications that span the full range of electronics – from a single-stage transistor amplifier to the most sophisticated microcontroller system. There really is something for everyone!

Each part includes a simple but useful practical test gear project that will build into a handy gadget that will either extend the features, ranges and usability of an existing item of test equipment or that will serve as a stand-alone instrument. We've kept the cost of these projects as low as possible, and most of them can be built for less than £10 (including components, enclosure and circuit board).

FREE COVER-MOUNTED CD-ROM

On the free cover-mounted CD-ROM you will find the software for the *PIC n' Mix* series of articles. Plus the full *Teach-In 2* book – *Using PIC Microcontrollers – A practical introduction* – in PDF format. Also included are Microchip's *MPLAB ICD 4 In-Circuit Debugger User's Guide*; *MPLAB PICkit 4 In-Circuit Debugger Quick Start Guide*; and *MPLAB PICkit4 Debugger User's Guide.*

ORDER YOUR COPY TODAY JUST CALL
01202 880299 OR VISIT www.electronpublishing.com

PIC n' Mix

Mike Hibbett's column for PIC project enlightenment and related topics

Part 3: PIC18F development board

We continue this month with the process of creating a development board for the PIC18F processor. In this and the following article we are going to cover the design of the board, ensuring it supports a number of different externally connected technologies: GPS location, digital compass, various display types, speech output, connectivity to a PC connection via USB, Wi-Fi communication, interfacing with Amazon Alexa, servomotor control and analogue input. The focus for this article series will be on making use of the processor, rather than diving into the finer details of internal processor peripherals – a hands-on, practical approach.

In the previous article we selected the processor we will be using – the PIC18F47K42 in a 40-pin DIL package. We chose that package because it provides lots of I/O signals yet is very easy to solder (via a 40-pin socket) making it easy to replace if damaged.

There is a lot of work to do before we jump in and start designing a PCB. Even before drawing the schematic, we need to be clear in our objectives for this board, writing down some requirements, and check if any of our items might clash with each other and (hopefully) find solutions to those issues. Let's start by writing down our key requirements:

- Easy to assemble – no tiny surface-mount soldered parts. We want this to be easily assembled, by any hobbyist
- Flexible – the ability to select which features are fitted at any one time, without major re-work
- Re-usable – easy to re-purpose to different projects
- PC serial communications interface using an on-board UART-to-USB converter chip, the MCP2221A-I/P
- Header connector for an ESP-01 Wi-Fi interface
- Header for a Micro SD Media card module
- Several three-pin headers for servomotors
- Header for a colour touchscreen LCD
- FET power switches for external device control
- PICkit 4 header for programming/debugging
- Header pins for external devices with power and I²C or SPI buses
- Two configurable op amps, for analogue input signal conditioning
- 32kHz crystal for very low-current operation
- Plenty of analogue input and digital I/O headers
- Single status LED, and a power LED
- Power input from a standard DC power brick.

That's quite a list, but many of these will be easy to implement, and simply remind us to include certain components in the schematic. Many requirements have a consequence – adding a 32kHz crystal, for example, means that two GPIO pins become unavailable. Fortunately, however, there are many GPIO pins, so that won't be an issue for us.

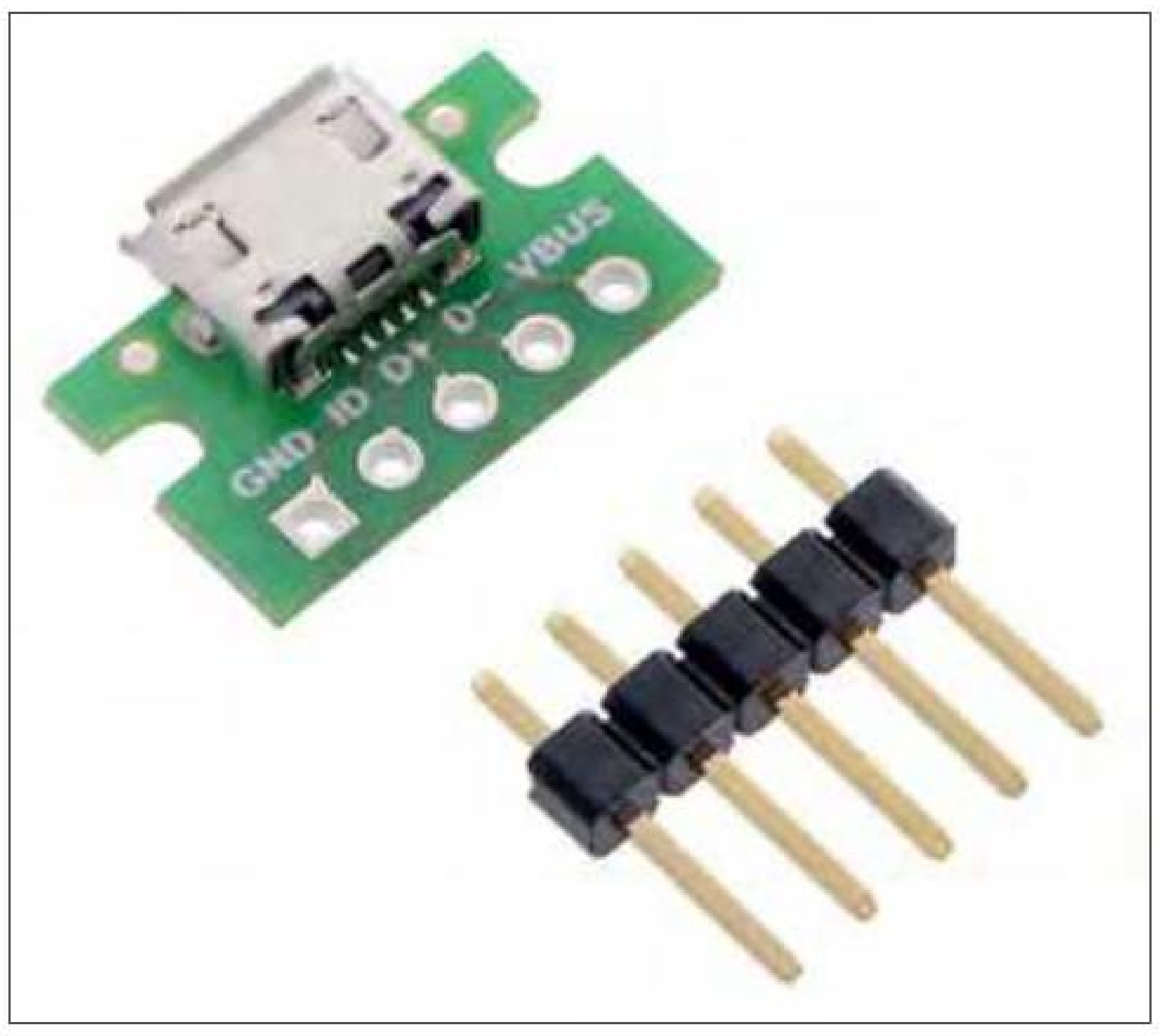


Fig.1. USB connector board.

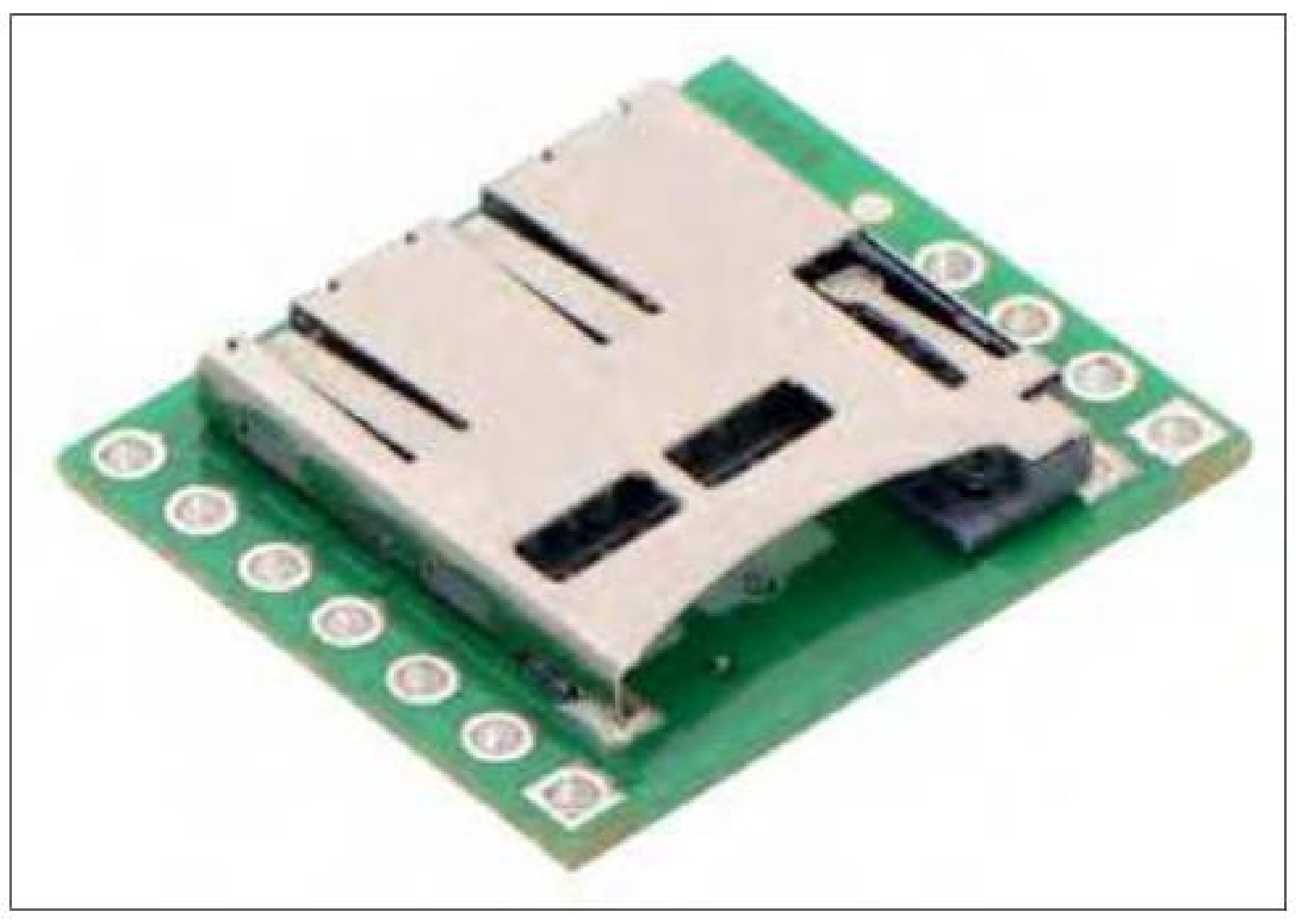


Fig.2. SD-Media connector board.

Simplifying soldering

The first requirement is driven by a desire to enable this board to be built by as many hobbyists as possible, avoiding the use of difficult-to-solder surface-mount parts. This is easy to achieve with most components, but the USB and SD Media connectors are only available in hard-to-solder formats. Thankfully, this has been recognised by a number of electronics suppliers who provide very simple and low-cost pre-assembled carrier boards, as shown in Fig.1 and Fig.2. We will also use a Wi-Fi module with a similar simple header interface (shown in Fig.3). These little boards enables us to avoid the difficulty of soldering ICs that are only available in surface-mount format. Another benefit of providing these as plug-in features is that they are optional – buy them only if you want to use them.

On our PCB, USB connectivity to a PC is provided by a special IC, the Microchip MCP2221A-I/P. Our processor does not have USB built in, and the choice to use an external IC rather than choose a processor with a USB peripheral is deliberate – in our experience the USB software libraries provided by Microchip (and other vendors) are very complicated to use. The MCP2221A provides a simple USB-to-UART conversion, so our processor we will connect the IC to one of our serial ports, and the USB cable will look to the PC like a standard serial port. This will make software development

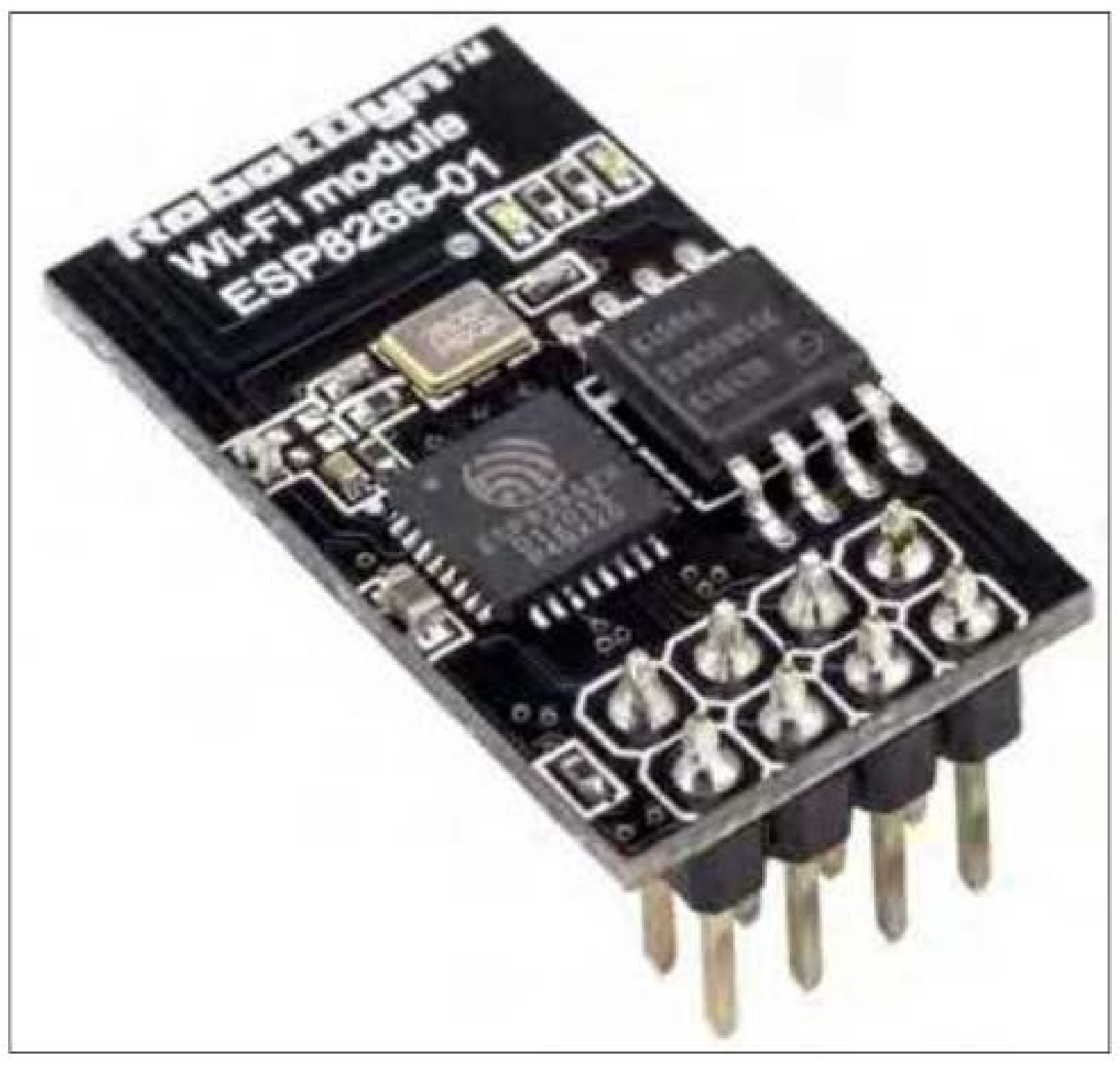


Fig.3. Wi-Fi interface board.

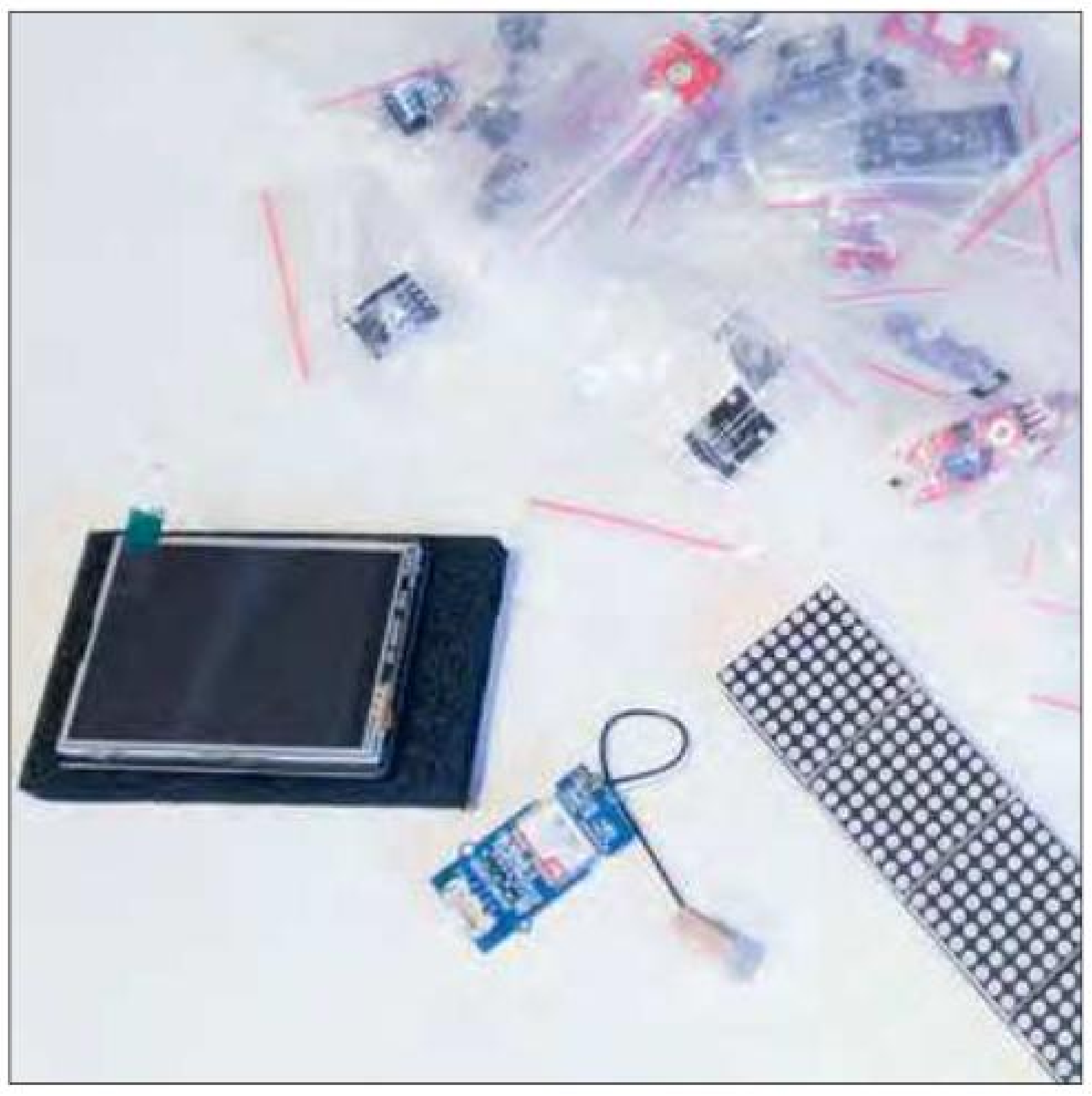


Fig.4. Plug-in option boards.

on both the PC and the processor simple. Plus, the MCP2221A can run at 3.3V or 5V, so it's very flexible.

An SD Media card communicates over an SPI interface, so we will route one of the SPI buses to it. SD Media cards operate at a voltage of 3.3V maximum, so bear that in mind if you are thinking of running your board at 5V. An external adaptor to level-shift the signals will be required if you want to use an SD Media card for data storage and run the board at 5V simultaneously. This will be true for the Wi-Fi interface too.

The downside of using standard components in a design like this is that the PCB will larger than the more usual Microchip development boards, which use tiny surface-mount components. This is not really a big issue – the board is for development purposes, so easy soldering and easy access to components is more important than size.

Plug-in options

Although this will be a general-purpose development board, we do have in mind some external devices that we would like to be able to connect to it. Fig.4 shows the current selection from our lab – An LED matrix, touch-input colour LCD screen, GPS module and a bag of over 30 different random sensors that were purchased cheaply on eBay. We will provide links for the parts used in upcoming articles, as we make use of them. Needless to say, these will just be examples, you are free to attach whatever you want to your board.

Wi-Fi interface

Choosing a Wi-Fi interface board presented some interesting challenges. The board has to be cheap, easy to purchase, easy to solder and easy to use. We settled on a relatively old device, the ESP-01, which is based on the ESP8266 processor. This is such a popular design that it has become available from dozens of different suppliers and is incredibly cheap – as low as £3 including delivery, if you do not mind the long delivery times from Asia. The module comes pre-loaded with software that is easy to use and communicates with our processor over one of the UART interfaces (we have two UARTs on our processor, so simultaneous Wi-Fi and USB connectivity will be possible.)

Power supply design

With those thoughts behind us, it's time to make our first key design decisions – how will we power the board, and how will we direct the processor's 36 I/O pins to on-board and external connections?

We'll start with the power supply design. So, first, we must decide at what voltage to run the board: 3.3V or 5V? These two common operating voltages have for years been a bone of contention for us, too often we have tried to pair a 5V external device with a 3.3V processor development board (servomotors with a Raspberry Pi for example) or a 3.3V module with a 5V processor board (a GPS module with an Arduino, for example.) It's been such an issue over the years that we have decided to design a board that can operate at either voltage, selected by a simple jumper.

The key ICs in our design have been selected as ones that can operate at either voltage, so we only need a power supply design that can be switched between the two voltages – and we can do that with an LM317 regulator and a few resistors. We will use a beefy TO220-style regulator, as powering some external devices, particularly servo motors, may bring our current-consumption requirements up to around 500mA or so. Plus, we make a mental note: leave space around the regulator for heatsinking, the LM317 is a linear regulator, and will get warm at these higher currents. We could have gone with a much more efficient switching-regulator design, but that would add complexity. We will typically be powering this board from a wall socket power supply, so power supply efficiency is not a design requirement.

Studying the datasheet of the LM317 we spot that is has a requirement for a minimum 3V input to output voltage differential, so to generate a 3.3V or 5V output will require an input power supply running at a minimum of 8V DC. 9V and 12V DC power supplies are common and cheap, so the regulator choice works well.

We will include a power supply isolation header pin, so 3.3V or 5V can be supplied to the board directly, if desired. This will enable the testing of low-power designs, or allow the addition of a more efficient power supply should you want to use a battery.

A barrel jack will be accommodated to support plugging in a standard DC power brick. There are several different standards for barrel jack connectors (different pin diameters) so we have to make a decision on which to go with – we chose the 2.5mm version, as that seemed to be the most common size in our lab. Header pins and solder pads will also be provided for power input, as these cost nothing but add flexibility to the choice of power connection.

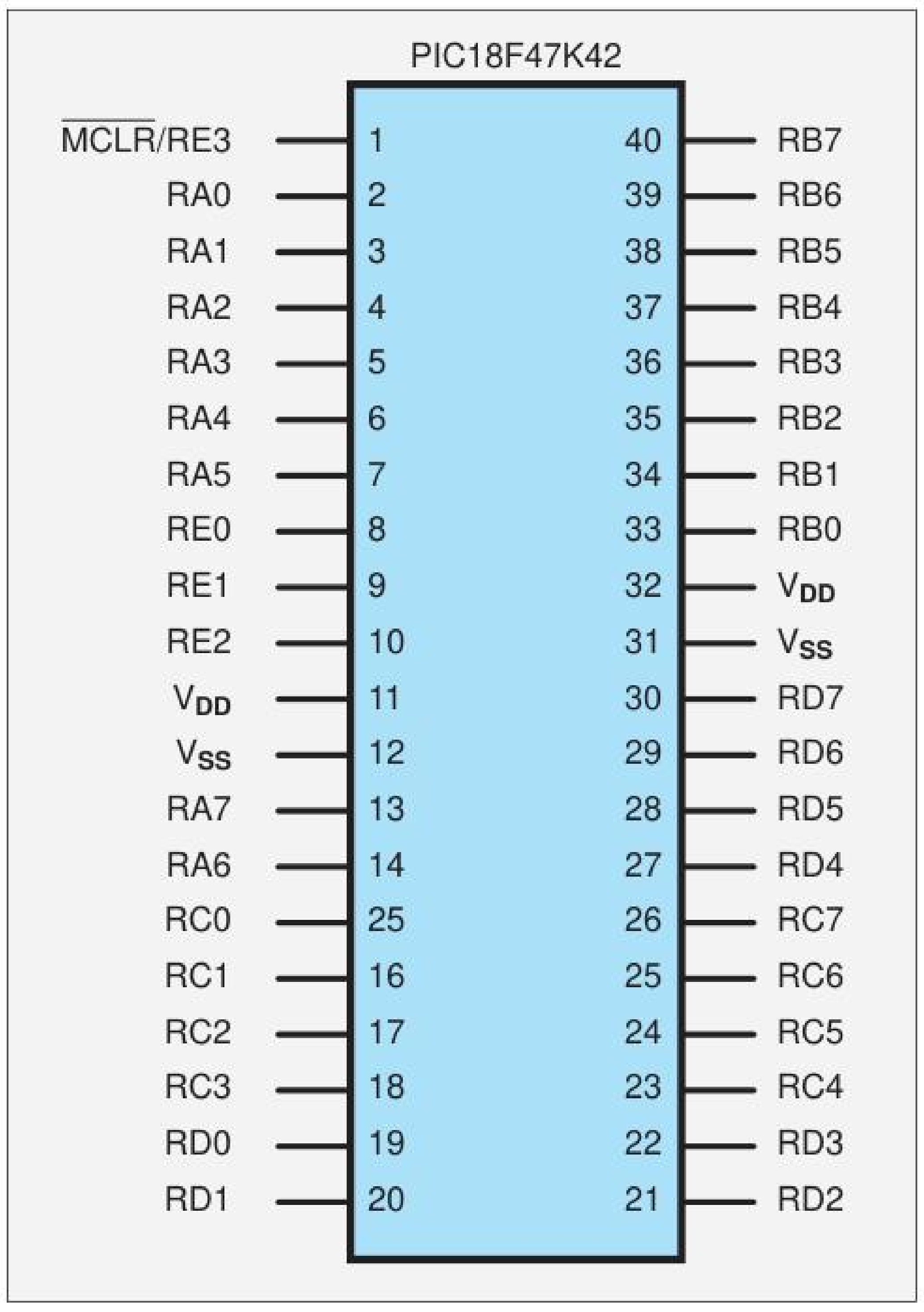


Fig.5. PIC18F47K42 processor pin-out.

External header pins

We have left the most complex design decision until last. The processor, shown in Fig.5, has 36 I/O pins – how will we connect these? Thankfully, the pins are highly configurable; internal peripherals such as SPI, I^{2}C and UART interfaces are all able to be programmed to appear on any pin from a selection of the 36 pins available. While this simplifies circuit design and PCB layout, it does add complexity during software design. With a general-purpose development board design, where we make all pins available, this task is very complicated – but that task can be simplified by writing up a spreadsheet for the pins, indicating where each one will go.

The approach taken here is that multiple headers are provided for generic SPI and I^{2}C devices, several servomotor headers (using the standard header layout) and specific headers for the Wi-Fi, USB and Micro SD Media boards. Two FET-controlled power output headers are provided, and then all I/O pins will go to 0.1-inch pitch header strips. Even the I/O signals that are optionally connected to other headers on the board will be routed to the generic I/O headers. Three I/O pins will not be used. PORTE.3 is shared with the master reset pin MCLR, which we will keep as a reset pin. PORTC.0 and PORTC.1 are multiplexed with the external crystal inputs, which will be taken up with our 32kHz crystal, for ultra-low power operation.

To finish off, two op amps will be implemented on the board, but left unconnected, with headers to allow user choice of where they connect.

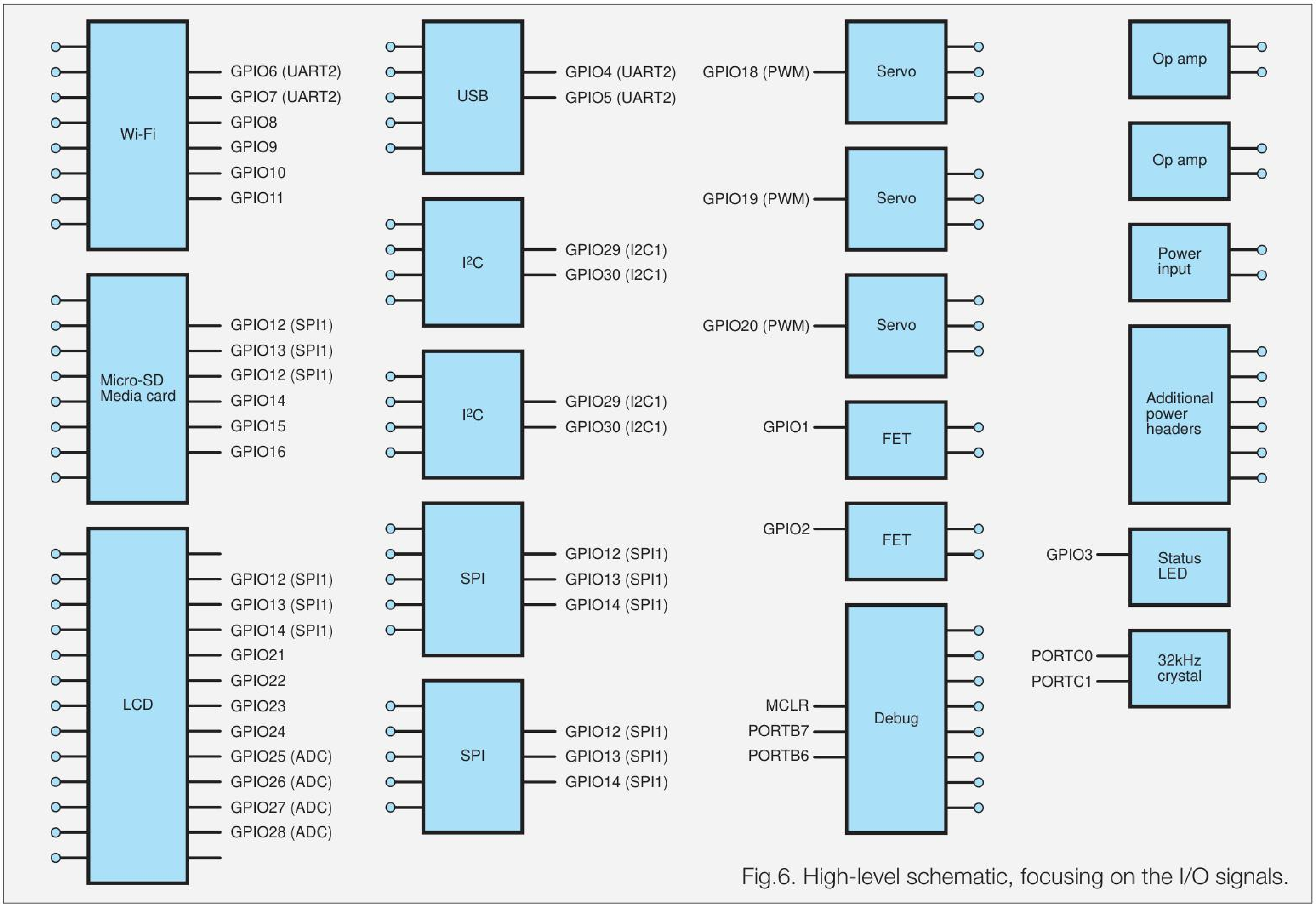

Wi-Fi
GPIO6 (UART2)
GPIO7 (UART2)
GPIO8
GPIO9
GPIO10
GPIO11
Micro-SD Media card
GPIO12 (SPI1)
GPIO13 (SPI1)
GPIO12 (SPI1)
GPIO14
GPIO15
GPIO16
LCD
GPIO12 (SPI1)
GPIO13 (SPI1)
GPIO14 (SPI1)
GPIO21
GPIO22
GPIO23
GPIO24
GPIO25 (ADC)
GPIO26 (ADC)
GPIO27 (ADC)
GPIO28 (ADC)
USB
GPIO4 (UART2)
GPIO5 (UART2)
I²C
GPIO29 (I2C1)
GPIO30 (I2C1)
I²C
GPIO29 (I2C1)
GPIO30 (I2C1)
SPI
GPIO12 (SPI1)
GPIO13 (SPI1)
GPIO14 (SPI1)
SPI
GPIO12 (SPI1)
GPIO13 (SPI1)
GPIO14 (SPI1)
GPIO18 (PWM)
Servo
GPIO19 (PWM)
Servo
GPIO20 (PWM)
Servo
GPIO1
FET
GPIO2
FET
MCLR
PORTB7
PORTB6
Debug
Op amp
Op amp
Power input
Additional power headers
GPIO3
Status LED
PORTC0
PORTC1
32kHz crystal

Fig.6. High-level schematic, focusing on the I/O signals.

A standard PICkit 4 8-pin header will be included for programming and debugging.

Working with the above specification, a high-level circuit design focusing on the I/O signals can be drawn, shown in Fig.6, which shows all the required signals, without specifically naming them. From this diagram we can now work through mapping processor pins to the diagram. This is a bit tedious and is best done by writing down a simple spreadsheet listing the pin numbers and their use, confirming for each pin that a chosen peripheral can be mapped to the desired pin. The resulting pin mapping is shown in Fig.7.

High-level design – done!

Having worked through those requirements, and followed the thought process discussed above, we have arrived at a clear high-level design. At this stage, we have no further questions to answer – the schematic can be drawn up, and a board layout created, with the risk of unexpected issues cropping up significantly reduced.

Finishing touches

Before getting to the PCB layout we note a few addition requirements that we want to capture.

Four holes in the PCB corners will be provided for the mounting of simple feet. 3.4mm-diameter holes will be used to allow the use of cheap M3 bolts to be used, but there will be space for off-the-shelf rubber feet.

Space will be left around the LM317 voltage regulator to allow for a small heatsink, should your attachments require large currents.

The silk screen of the PCB will provide clear pin labels for each header pin. This is an important requirement and will increase the size of the PCB slightly as space beside the header pins will be needed to allow the text to be visible.

Although it will not impact the PCB design, it's worth noting that all ICs (the processor, the USB interface IC and the op amp) will be fitting using sockets. Although this adds a few pounds to the cost, it will make the board easier to repair should accidents happen. You are of course free to ignore this advice!

That's all the hard work completed – now to the PCB design itself, which we will pick up in the next article.

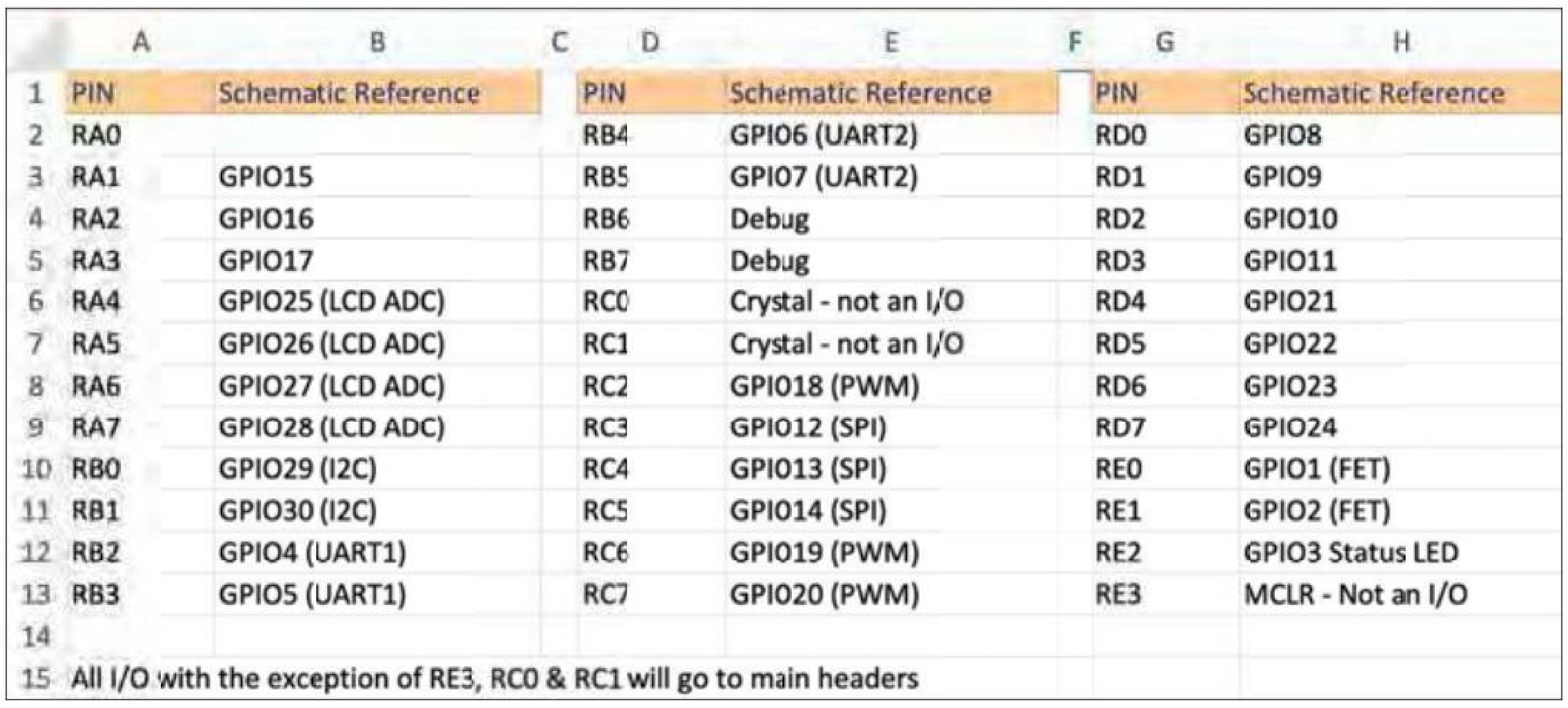
References

Micro SD-card holder
www.pololu.com/product/2597

Wi-Fi interface
www.addicore.com/ESP8266-ESP-01-p/130.htm

USB Micro-B connector
www.pololu.com/product/2586

LED Display
www.ebay.co.uk/itm/191736670249



	A	B	C	D	E	F	G	H
1	PIN	Schematic Reference		PIN	Schematic Reference		PIN	Schematic Reference
2	RA0			RB4	GPIO6 (UART2)		RD0	GPIO8
3	RA1	GPIO15		RB5	GPIO7 (UART2)		RD1	GPIO9
4	RA2	GPIO16		RB6	Debug		RD2	GPIO10
5	RA3	GPIO17		RB7	Debug		RD3	GPIO11
6	RA4	GPIO25 (LCD ADC)		RC0	Crystal - not an I/O		RD4	GPIO21
7	RA5	GPIO26 (LCD ADC)		RC1	Crystal - not an I/O		RD5	GPIO22
8	RA6	GPIO27 (LCD ADC)		RC2	GPIO18 (PWM)		RD6	GPIO23
9	RA7	GPIO28 (LCD ADC)		RC3	GPIO12 (SPI)		RD7	GPIO24
10	RB0	GPIO29 (I2C)		RC4	GPIO13 (SPI)		RE0	GPIO1 (FET)
11	RB1	GPIO30 (I2C)		RC5	GPIO14 (SPI)		RE1	GPIO2 (FET)
12	RB2	GPIO4 (UART1)		RC6	GPIO19 (PWM)		RE2	GPIO3 Status LED
13	RB3	GPIO5 (UART1)		RC7	GPIO20 (PWM)		RE3	MCLR - Not an I/O
14								
15	All I/O with the exception of RE3, RC0 & RC1 will go to main headers							

Fig.7. I/O pin mapping.

AUDIO OUT

By Jake Rothman

Theremin Audio Amplifier – Part 3

Last month, we completed a silicon-transistor-based amplifier fo the *PE Theremin*, but I concluded the article with a promise to look at a germanium-transistor version. This actually turned into quite a fascinating project, as you will see.

Lost values

When I get a break from eyeball-stressing SMT (surface-mount technology) industrial jobs, it's a relief to get back to some old-fashioned electronics. I recently got really fed up with unmarked SMT components. It started with capacitors, but now even resistors are being supplied unmarked, as shown in Fig.1. This makes fault finding on pre-production prototypes very difficult. When the unit fails a few years later it can't be fixed without full service data, and who supplies that today?

Nice old junk

Sifting through some musty boxes (Fig.2) courtesy of yet another lock-down delivery of components rendered obsolete by RoHS directives, my childhood electronics excitement came back. Behold! – there were unopened packs of 1974 Siemens germanium transistors.

There was also a bundle of Siemens books, including one called *Design Examples of Semiconductor Circuits*, from 1969. The section titled, 'Audio-frequency Amplifiers' on p.7 compared the use of germanium output transistors with the then relatively new silicon planar transistors. It said the germanium types were superior for low voltage, which is to be expected, since their turn-on voltage is around 0.1V as opposed to silicon's 0.6V. What intrigued me, however, was the statement that they had better linearity, which should mean lower distortion. The amplifier designer and Bradford University academic Arthur Bailey also said the same in his *Wireless World* article, 'High Performance Transistor Amplifier' (see p.543, November 1966). I'm always a bit wary about 'linearity' and other spurious audio claims, but there is only one way to be sure – build and compare.

On pages 14 and 15, the Siemens book had two almost identical four-transistor amplifier circuits for a 9V supply and 8Ω speakers, see Fig.3. The main difference was one used the germanium output transistors (AC153K and AC176K) in the box and the other used silicon (BC140 and BC160). Biasing requirements accounted for the other circuit differences. The claimed output power was 1.1W for the germanium and 0.7W for the silicon.

This sort of germanium amplifier was used in Robert's Radios, such as the R505, until the early 1970s. They still sound good today – and they don't eat batteries. Maybe their designers were onto something.

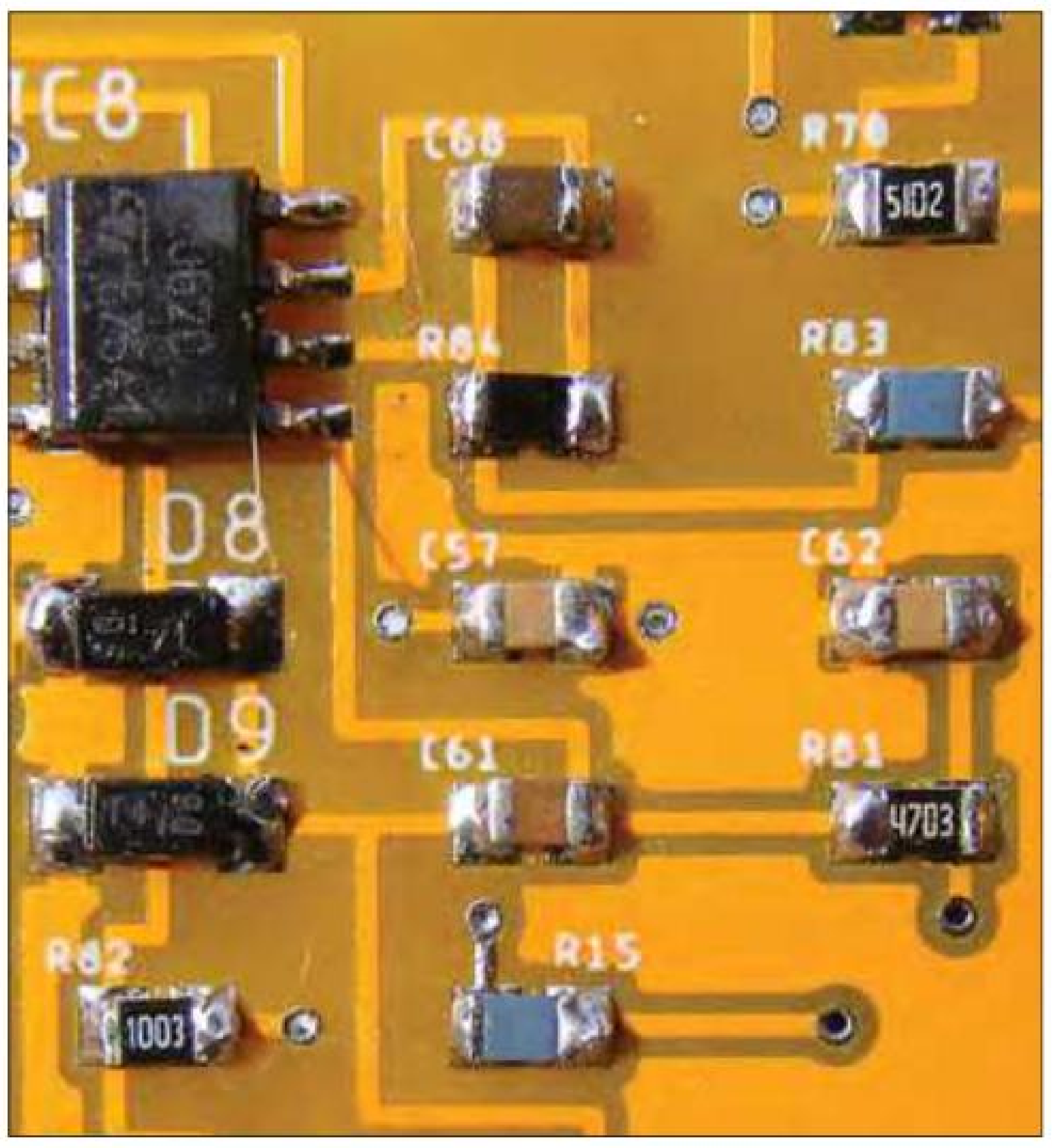


Fig.1. It started with ceramic chip capacitors. Now, even some resistors have no markings, such as these Vishay CRCW080522K0FKEA resistors from RS. Just black and blue rectangles (R84 and R15). This is what I have to put up with in my professional life – I am a human, not a pick-and-place machine!

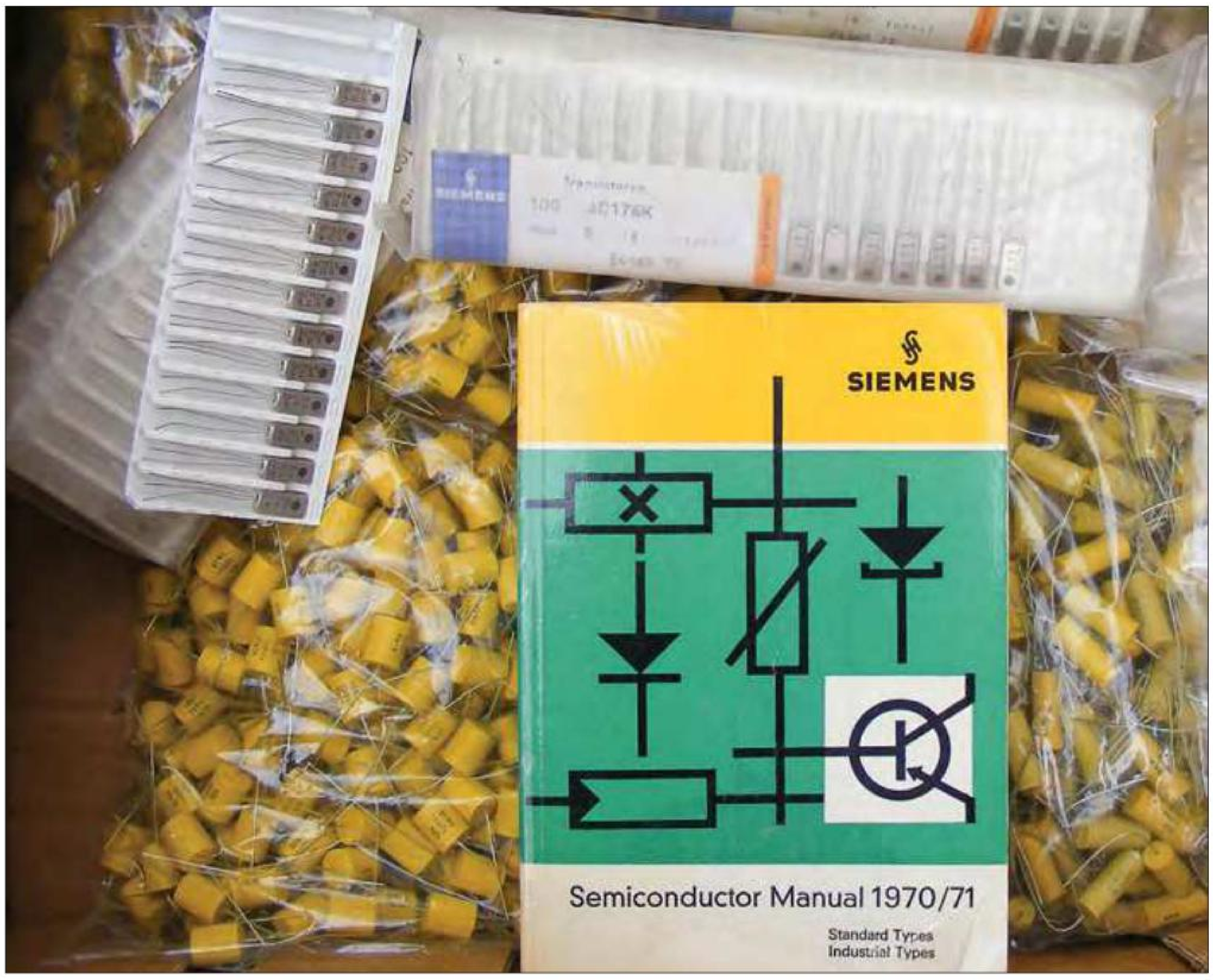


Fig.2. A good dollop of old 'proper' components with wires and markings. Siemens germanium output transistors and Philips 'Minipoco' 1% tolerance lead-foil polystyrene capacitors. Still 1% accurate after 35 years.

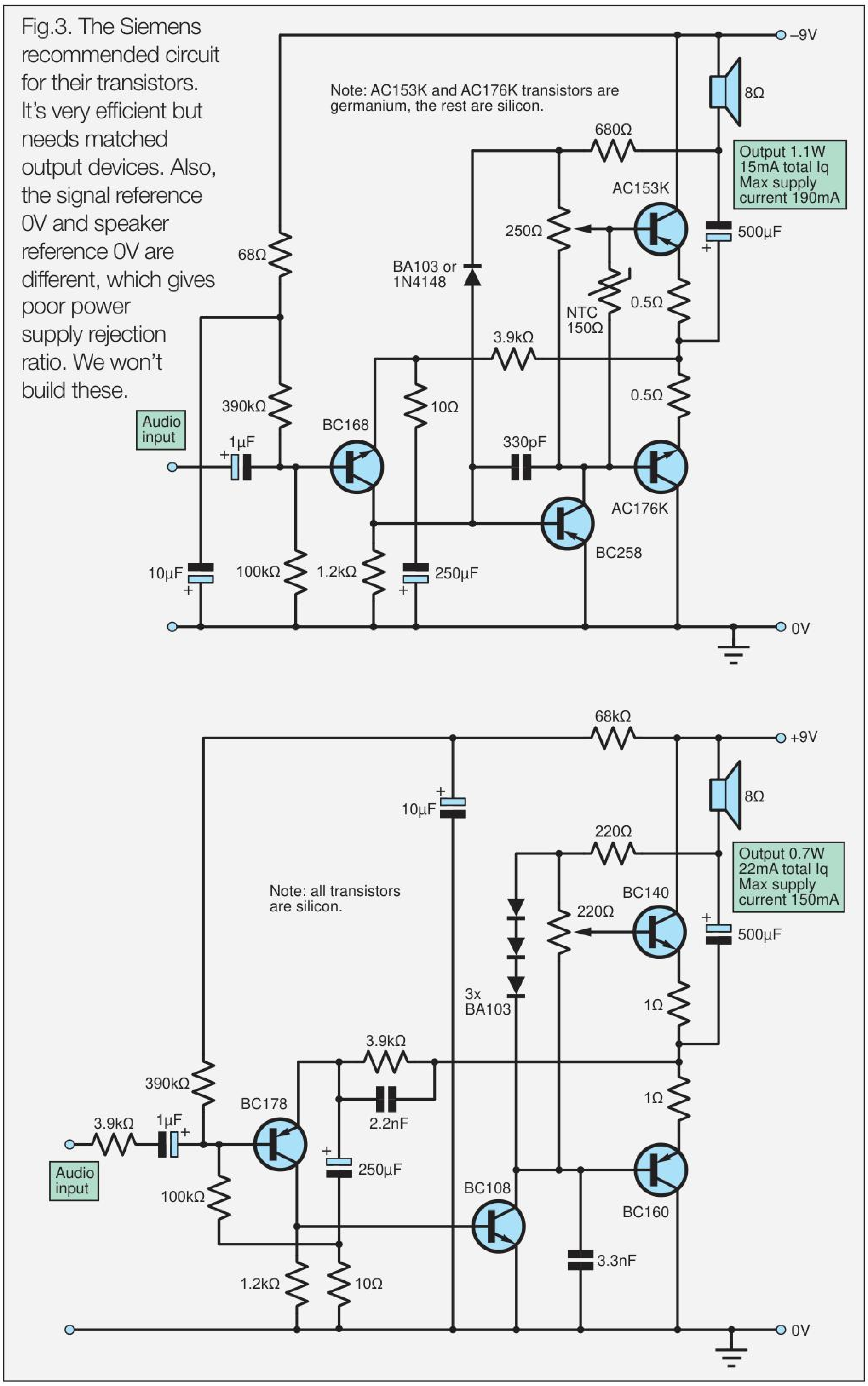

Note: AC153K and AC176K transistors are germanium, the rest are silicon.
–9V
8Ω
680Ω
Output 1.1W
15mA total Iq
Max supply current 190mA
AC153K
250Ω
500µF
68Ω
BA103 or 1N4148
NTC 150Ω
0.5Ω
3.9kΩ
0.5Ω
390kΩ
10Ω
Audio input
1µF
BC168
330pF
AC176K
BC258
10µF
100kΩ
1.2kΩ
250µF
0V
68kΩ
+9V
10µF
8Ω
220Ω
Output 0.7W
22mA total Iq
Max supply current 150mA
Note: all transistors are silicon.
BC140
220Ω
500µF
3x BA103
1Ω
3.9kΩ
390kΩ
1Ω
BC178
2.2nF
3.9kΩ
1µF
Audio input
250µF
100kΩ
BC108
BC160
3.3nF
1.2kΩ
10Ω
0V

Fig.3. The Siemens recommended circuit for their transistors. It's very efficient but needs matched output devices. Also, the signal reference 0V and speaker reference 0V are different, which gives poor power supply rejection ratio. We won't build these.

Comparison circuit

To perform a practical germanium vs silicon comparison, I decided to use the *PE Theremin Amplifier* circuit and PCB from November/December 2020's *Audio Out*. I installed transistor sockets on the board – always a good idea when trying unusual parts. I used cut strips of SIL turned-pin IC sockets for this (see Fig.4), since real transistor sockets are expensive. I decided to go for a standard 9V, 8Ω set-up to see if I could replicate the Siemens amplifiers' results.

There is no point in trying to go for a micro-power design with germanium, since their high leakage currents prohibit this. When experimenting with germanium transistors it is important to check the leakage current (I_{cbo}) of old stock devices. This is the current that flows between collector and emitter with the base open-circuit. I use a Peak DCA75 Analyser for this (Fig.5), but a 5V power source and multimeter, as shown in Fig.6, will also do the job.

Power supply

Since germanium NPN transistors are rare, the circuit was inverted to use a negative power rail and positive earth. This way only one NPN transistor was needed, the rest are PNP devices, normal

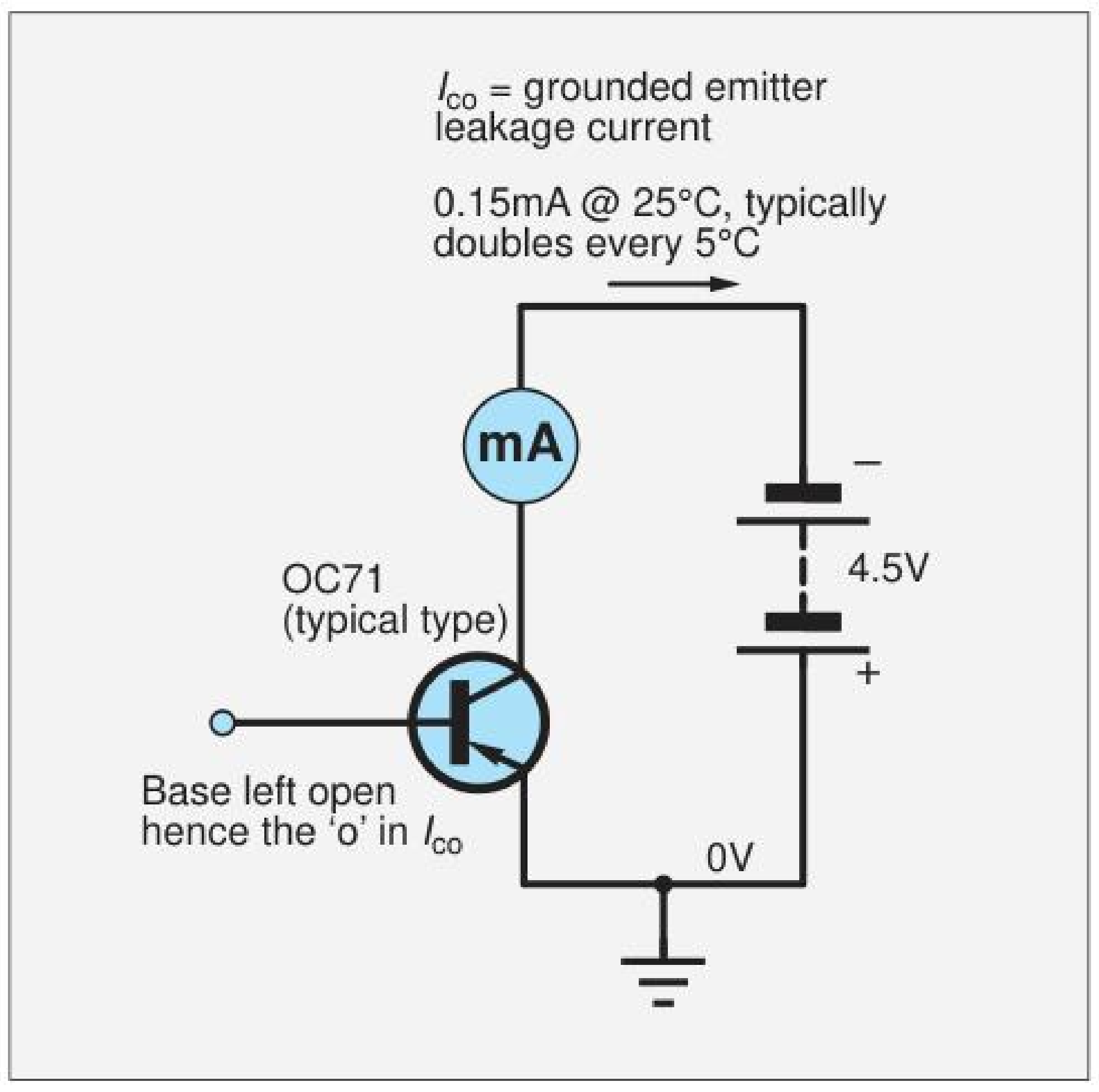

I_{co} = grounded emitter leakage current
0.15mA @ 25°C, typically doubles every 5°C
mA
–
4.5V
+
OC71 (typical type)
Base left open hence the 'o' in I_{co}
0V

Fig.6. Circuit for checking leakage current.

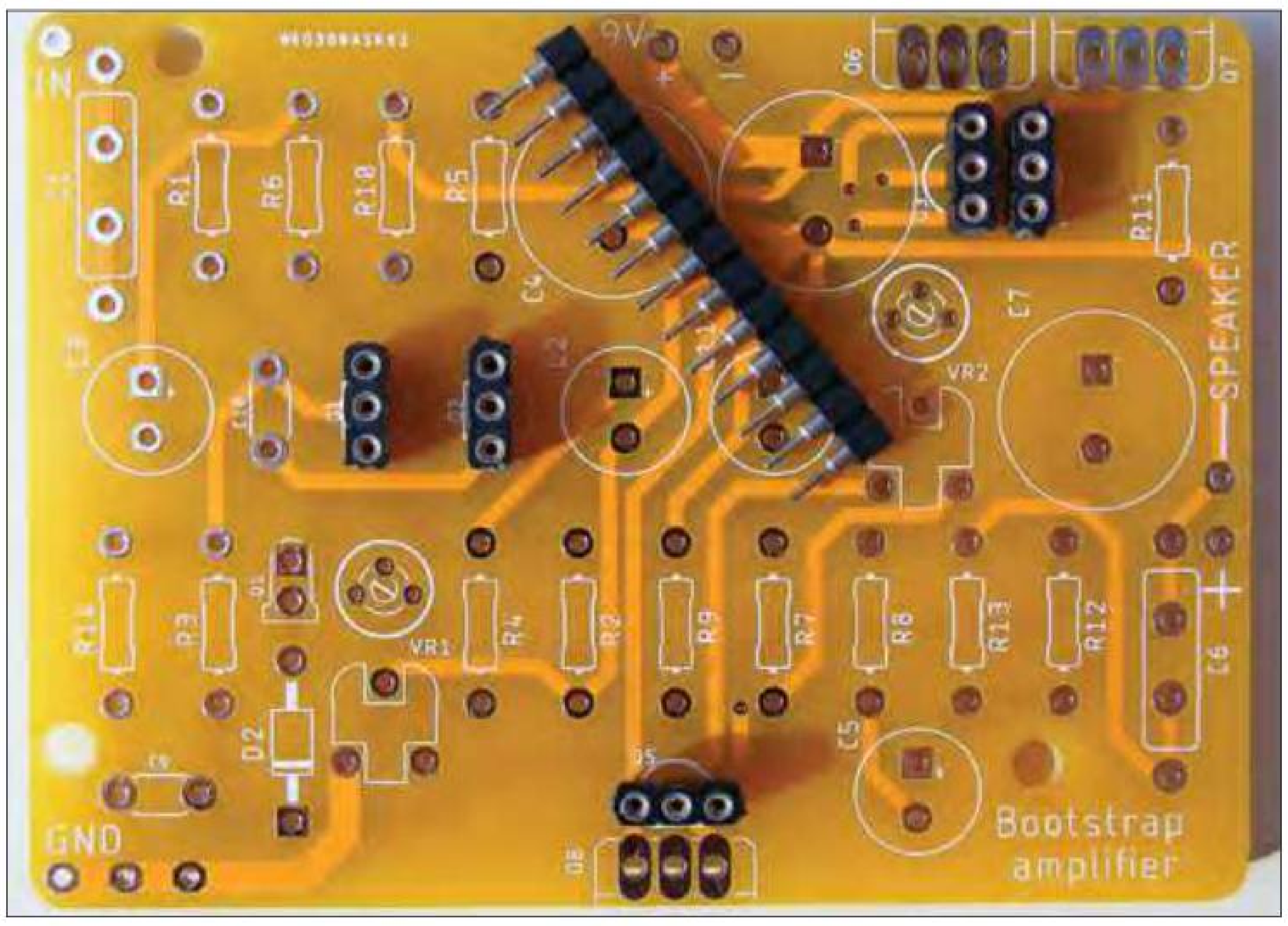

IN
9V
R1
R6
R10
R5
R11
SPEAKER
C7
VR2
VR1
R3
R4
R2
R9
R7
R8
R13
R12
C6
C5
D2
GND
Bootstrap amplifier

Fig.4. When trying out unusual transistors, use SIL sockets for easy substitution during R&D.

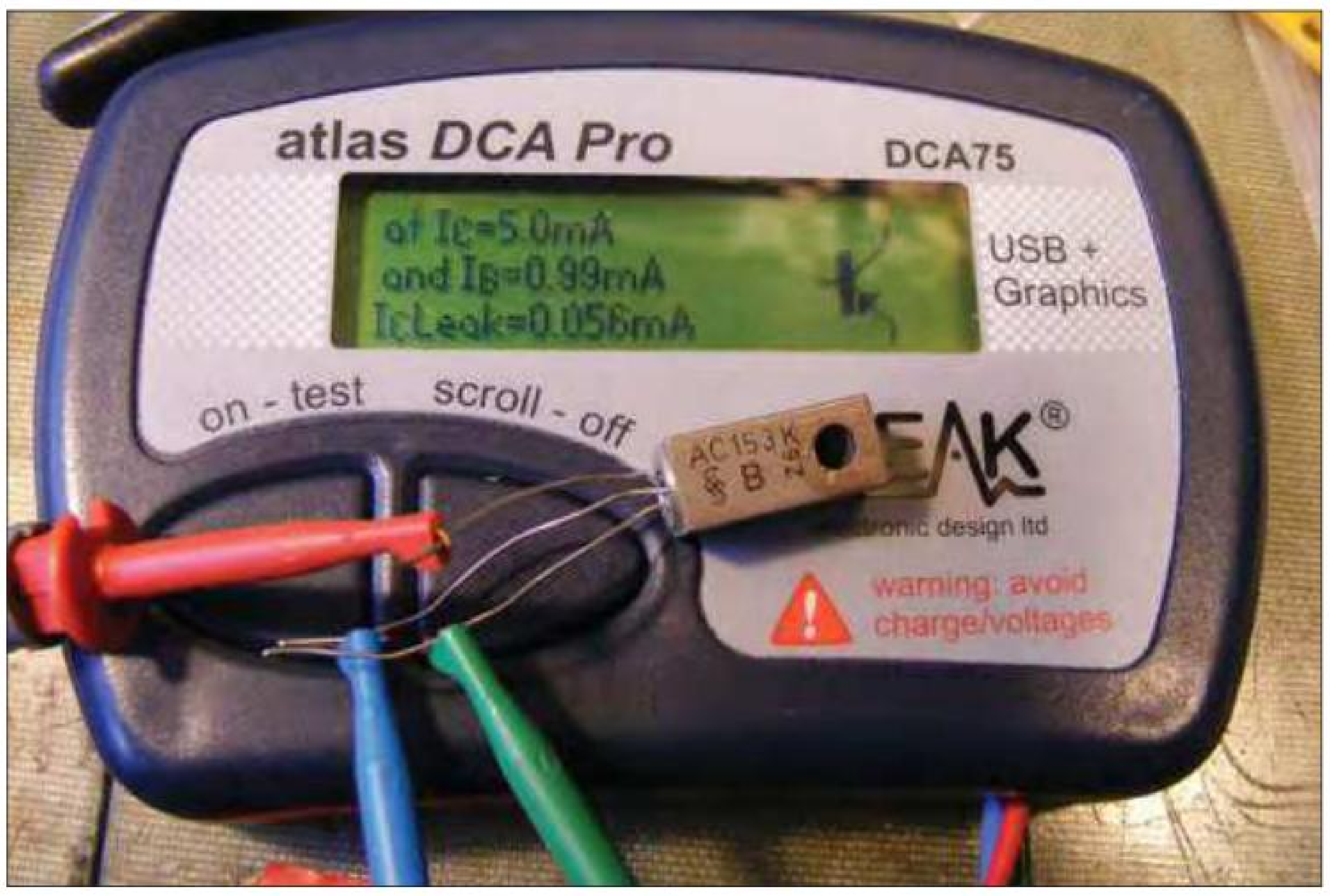

atlas DCA Pro
DCA75
of Ic=5.0mA
and Ib=0.99mA
IcLeak=0.056mA
USB + Graphics
on - test
scroll - off
AC153K
PEAK
electronic design ltd
warning: avoid charge/voltages

Fig.5. It is important to check the leakage currents of germanium transistors. Anything above 0.25mA is suspect, except for power types, where the limit is 2mA.

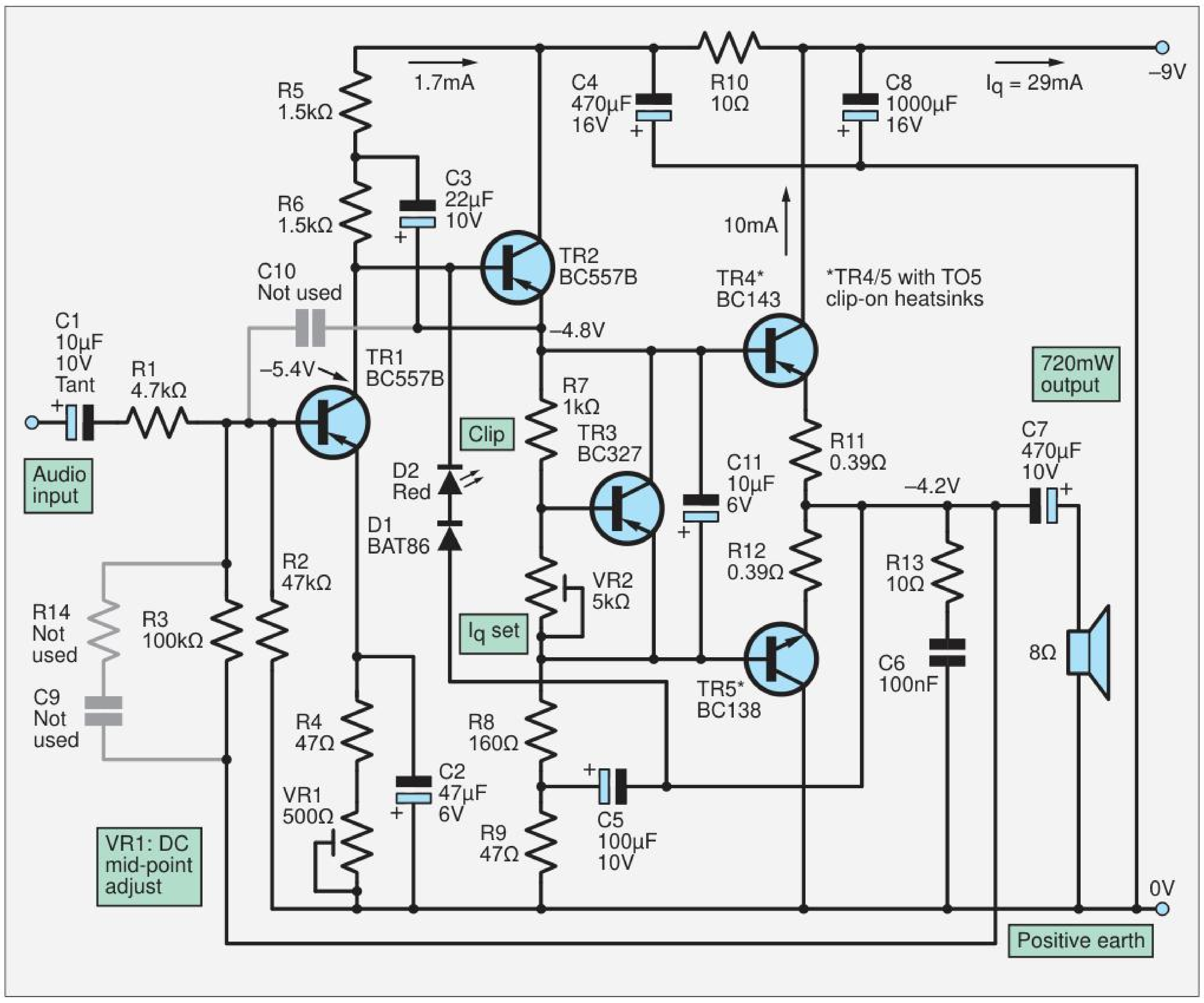

R5
1.5kΩ
1.7mA
C4
470µF
16V
R10
10Ω
C8
1000µF
16V
Iq = 29mA
–9V
R6
1.5kΩ
C3
22µF
10V
10mA
TR2
BC557B
TR4*
BC143
*TR4/5 with TO5
clip-on heatsinks
C10
Not used
–4.8V
C1
10µF
10V
Tant
R1
4.7kΩ
–5.4V
TR1
BC557B
R7
1kΩ
720mW
output
Clip
TR3
BC327
R11
0.39Ω
C7
470µF
10V
Audio
input
D2
Red
C11
10µF
6V
–4.2V
D1
BAT86
R12
0.39Ω
R13
10Ω
R2
47kΩ
VR2
5kΩ
R14
Not
used
R3
100kΩ
Iq set
8Ω
C6
100nF
TR5*
BC138
C9
Not
used
R4
47Ω
R8
160Ω
C2
47µF
6V
VR1
500Ω
C5
100µF
10V
R9
47Ω
VR1: DC
mid-point
adjust
0V
Positive earth

Fig.7. To start the development of the germanium amplifier, a silicon amplifier was developed using a negative power rail with positive earth. This way, only one NPN device was needed, with the rest being PNP types, paving the way for step-by-step substitution with germanium devices.

for germanium. Of course, when doing this 'mirroring', all the electrolytic capacitor and diodes have to be reversed too. This is no problem with the silicon circuit, since the transistors cost peanuts, whatever the polarity. Later I'll use this positive-earth amplifier design with some germanium fuzz circuits and Mullard LP1171/69 radio modules, as used in the Robert's R600. (If you fancy creating your own 'Robert's Radio sound' then I've even got some old alnico magnet Celestion 6×4-inch speakers for sale).

Germanium gestation

The first task is to get the silicon version working. That way any transistor blow-ups will cost 5p rather than possibly £1. To do this, we'll take the original circuit (Fig.9, p.65, *PE* November 2020) and scale up the currents by a factor of three for the lower speaker impedance of 8Ω, rather than the original 25Ω. This entails dividing most resistor values by three. Capacitors will have to be increased by three to take into account the lower impedances. Finally, the power rails and all other polarised components are flipped. There are then resistor tweaks to bring the biasing into line. This new circuit is shown in Fig.7.

An interesting high-frequency instability occurred where the frequency of oscillation was so high my 40MHz 'scope couldn't see it, but it was enough to completely mess-up the DC conditions and drive one output transistor to partly cut off. The clue was when I put my finger near R6 and the problem just cured itself, a sign of VHF oscillations. Ironically, removing stabilising capacitor C10 fixed it. When the new silicon circuit was rebuilt using a standard positive rail (Fig.8) the instability vanished and there was no problem with C10. I have heard that there is more likely to be problems with silicon PNPs; they have more junction capacitance modulation (Early effect) than NPNs.

The germanium transistors were then gradually plugged in, checking along the way with a multimeter, 'scope and signal generator. The only significant changes were to the DC biasing, with R2 being changed from 47kΩ to 15kΩ. Of course, the output current bias (V_{be} multiplier) transistor has to be germanium if the output devices are. This can be an AC153 or a special low-voltage transistor designed/selected especially for the job, such as an AC169. Using an NKT214 didn't work because the I_q could not be turned down to zero. Do note that some AC169s only have two wires, they are essentially just a 0.13V diode. The metal-cased three-wire ones have to be checked on a transistor tester to determine the connections.

Germanium-alloy junction transistors are generally ten-times slower than silicon planar types, giving inferior square-wave and high-frequency response. With TR1 and TR2 (in Fig.9) being 2MHz-F_t (frequency at which gain falls to unity) NKT214 audio types, the –3dB point was about 15kHz, fine for guitar and AM radio. The square-wave response at 1kHz 2V_{pk-pk} output had overshoot on it from TR2. Connecting a 33pF capacitor (C10) removed this. The high-frequency response was brought up to Hi-Fi (40kHz bandwidth) standards by putting in germanium 2SA12 RF transistors, as used in AM radio oscillators and leaving C10 out. These have an F_t of around 10MHz. Finally, we had a germanium amplifier with good performance. The circuit is shown in Fig.9.

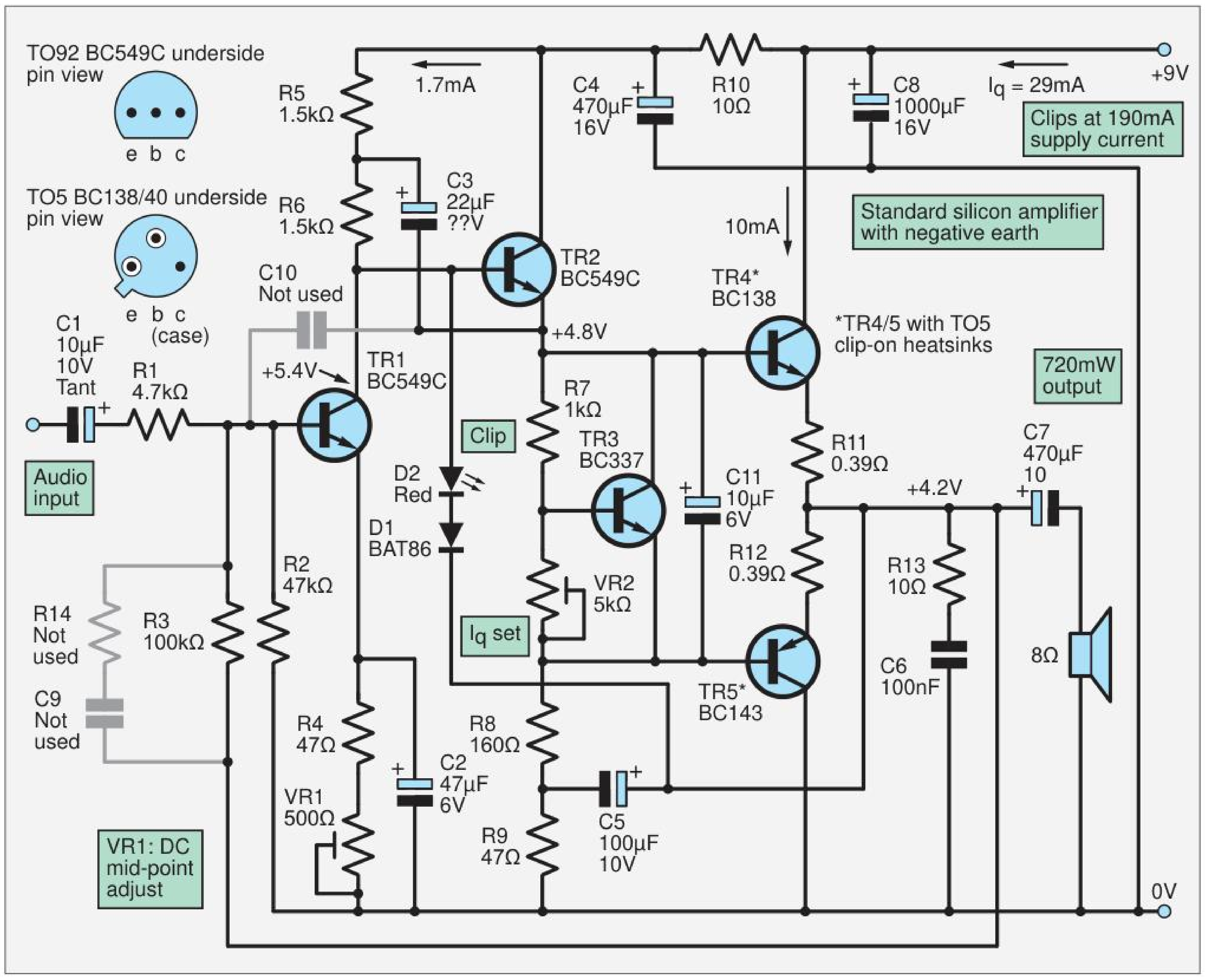

TO92 BC549C underside
pin view
e b c
TO5 BC138/40 underside
pin view
e b c
(case)
R5
1.5kΩ
1.7mA
C4
470µF
16V
R10
10Ω
C8
1000µF
16V
Iq = 29mA
+9V
Clips at 190mA
supply current
R6
1.5kΩ
C3
22µF
??V
Standard silicon amplifier
with negative earth
10mA
TR2
BC549C
TR4*
BC138
C10
Not used
+4.8V
*TR4/5 with TO5
clip-on heatsinks
C1
10µF
10V
Tant
R1
4.7kΩ
+5.4V
TR1
BC549C
720mW
output
R7
1kΩ
Clip
TR3
BC337
R11
0.39Ω
C7
470µF
10
Audio
input
D2
Red
C11
10µF
6V
+4.2V
D1
BAT86
R12
0.39Ω
R13
10Ω
R2
47kΩ
VR2
5kΩ
R14
Not
used
R3
100kΩ
Iq set
8Ω
C6
100nF
TR5*
BC143
C9
Not
used
R4
47Ω
R8
160Ω
C2
47µF
6V
VR1
500Ω
C5
100µF
10V
R9
47Ω
VR1: DC
mid-point
adjust
0V

Fig.8. The silicon circuit with standard negative earth, a useful medium-power amplifier.

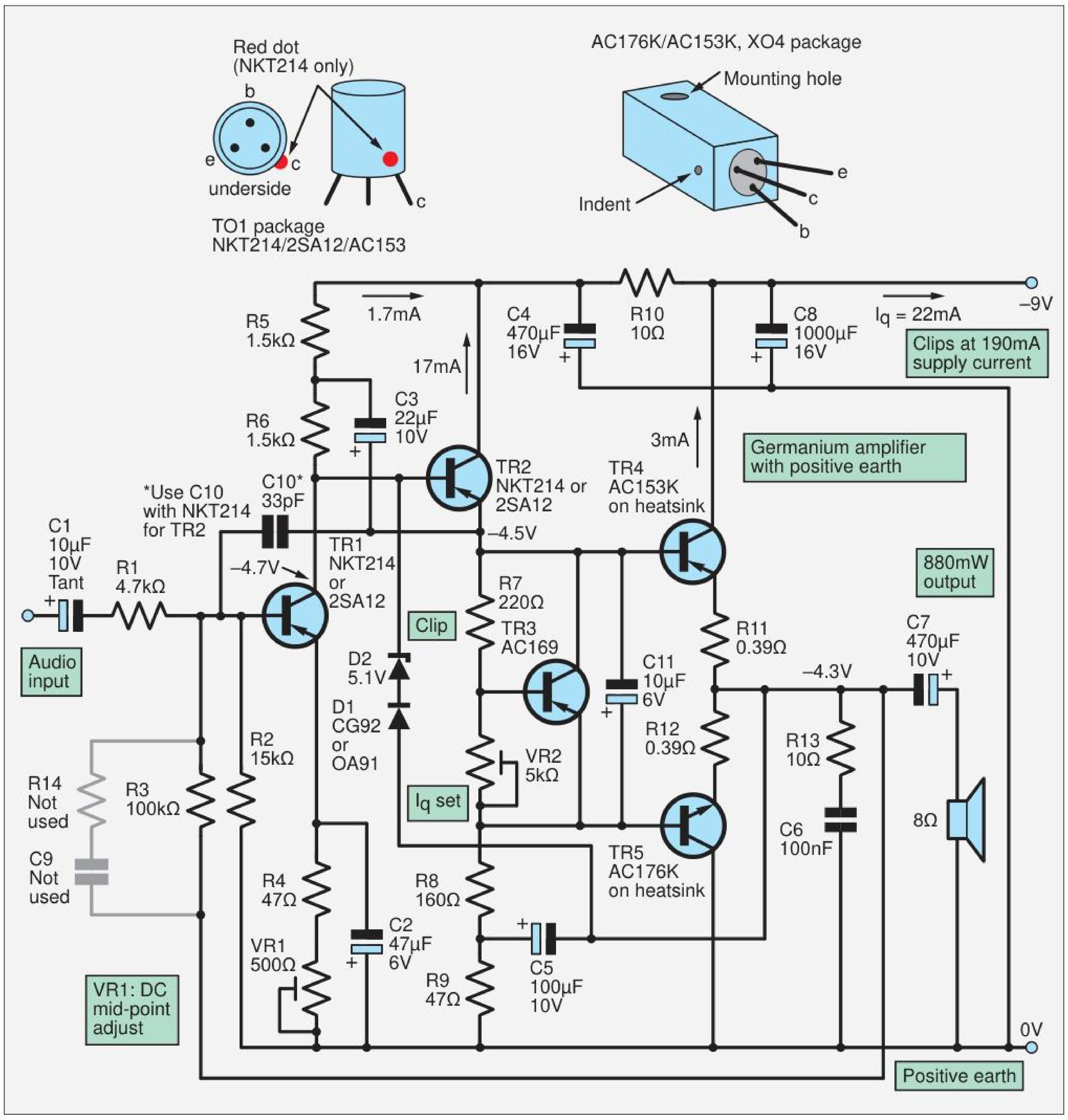

Red dot
(NKT214 only)
b
e
c
underside
c
TO1 package
NKT214/2SA12/AC153
AC176K/AC153K, XO4 package
Mounting hole
Indent
e
c
b
R5
1.5kΩ
1.7mA
17mA
C4
470µF
16V
R10
10Ω
C8
1000µF
16V
Iq = 22mA
–9V
Clips at 190mA supply current
R6
1.5kΩ
C3
22µF
10V
3mA
Germanium amplifier with positive earth
*Use C10 with NKT214 for TR2
C10*
33pF
TR2
NKT214 or
2SA12
TR4
AC153K
on heatsink
C1
10µF
10V
Tant
R1
4.7kΩ
–4.7V
TR1
NKT214
or
2SA12
–4.5V
880mW output
R7
220Ω
Clip
TR3
AC169
R11
0.39Ω
C7
470µF
10V
Audio input
D2
5.1V
C11
10µF
6V
–4.3V
D1
CG92
or
OA91
R12
0.39Ω
R2
15kΩ
VR2
5kΩ
R13
10Ω
R14
Not
used
R3
100kΩ
Iq set
8Ω
C6
100nF
C9
Not
used
TR5
AC176K
on heatsink
R4
47Ω
R8
160Ω
C2
47µF
6V
C5
100µF
10V
VR1
500Ω
R9
47Ω
VR1: DC mid-point adjust
0V
Positive earth

Fig.9. The final circuit using germanium transistors.

Output transistor selection

For the silicon amplifier, standard TO5 metal-can devices were used; in this case BC138 and BC143 – just because they were in the drawer. They can comfortably deliver output powers up to 2W with standard clip-on heatsinks. The germanium equivalents are the AC153K (PNP) and AC176K (NPN) shown in Fig10. The type with the K suffix was used because they have a convenient hole to mount on a metal bracket for heatsinking. The XO4 and TO1 germanium cases are electrically isolated, the heat passing from the junction to the case via a white paste filling of aluminium oxide and silicone grease. There are few germanium complementary pairs available. Alternatives are AC128/176 and AC188/187. For higher powers up to 6W the AD161/162 are often used (Fig.11). These TO66 devices are the germanium equivalent of the silicon TO126 BD135 and BD136 types.

Next month

That's all for this month. In *Part 4*, we'll finish with the component options and examine the frequency responses.

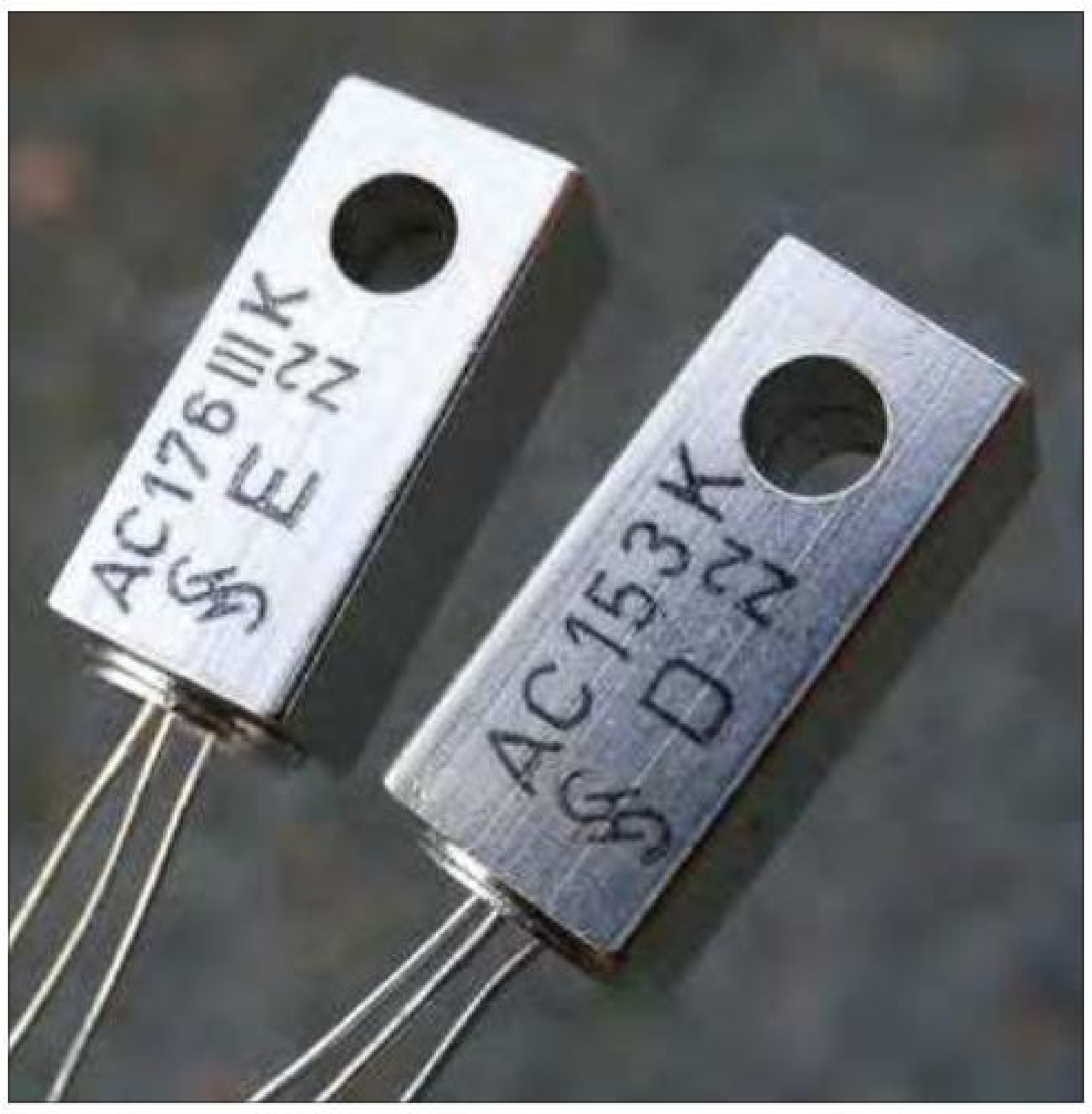

AC176 III K
ZZ
E
AC153K
ZZ
D

Fig.10. The Siemens AC176K and AC153K complementary output transistors. These are still used today in the very expensive EMS VCS3 synthesiser for their 'musical' sound.

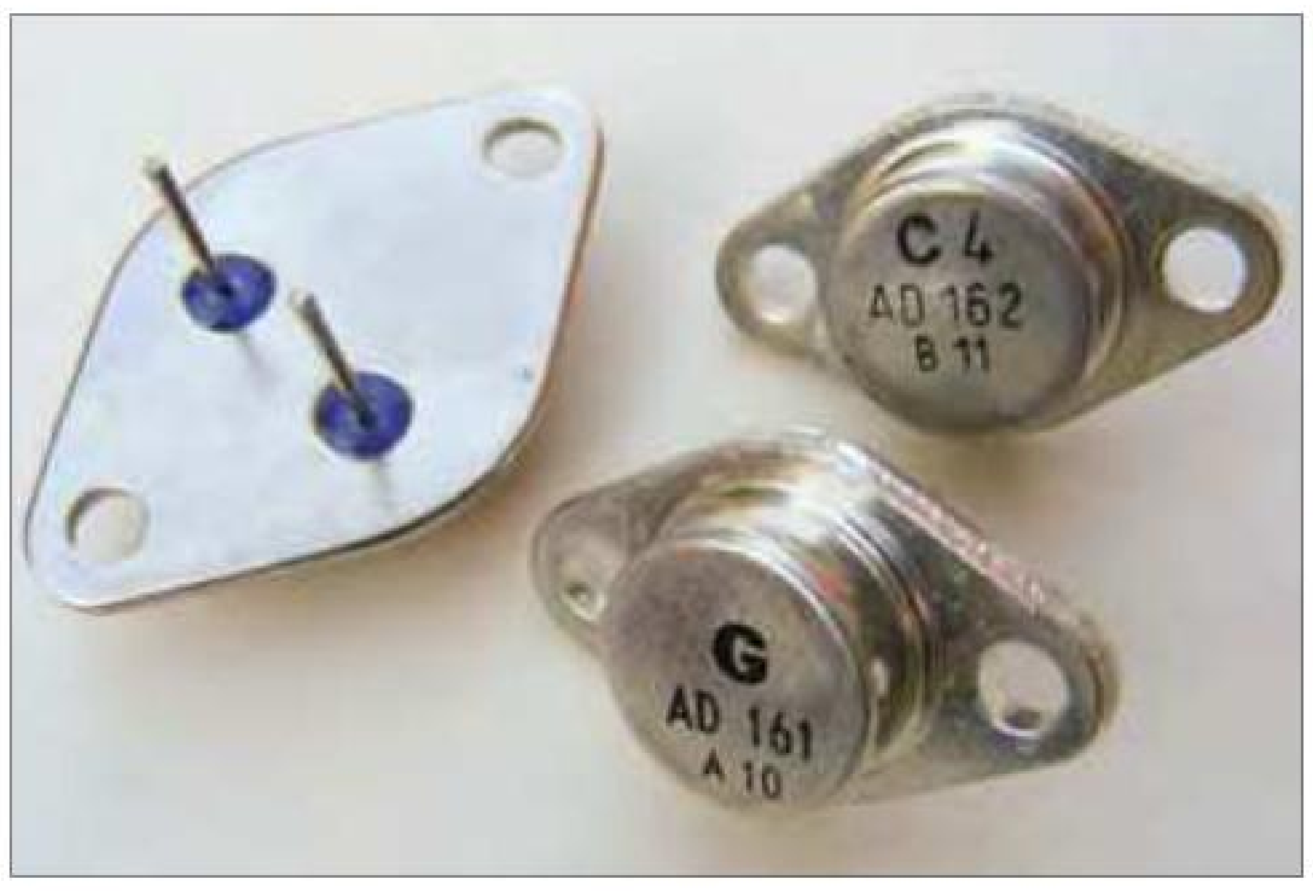

C4
AD162
B11
G
AD 161
A10

Fig.11. The Mullard AD161/2 output transistors (right). Very popular in the 1970s. Robert's used them in their RM50 table radio along with the Celestion speaker in Fig.18. Note their small TO66 case compared to the standard TO3 (left).

PoLabs

For more information or software download please visit

www.poscope.com/epe

PoKeys
Connect, Control

- USB
- Ethernet
- Web server
- Modbus
- CNC (Mach3/4)
- IO
- PWM
- Encoders
- LCD
- Analog inputs
- Compact PLC

Stepper motor drivers
Drive

- up to 256 microsteps
- 50 V / 6 A
- USB configuration
- Isolated
- up to 32 microsteps
- 30 V / 2.5 A

PoScope Mega1+
PoScope Mega50
Measure

- up to 50MS/s
- resolution up to 12bit
- Lowest power consumption
- Smallest and lightest
- 7 in 1: Oscilloscope, FFT, X/Y, Recorder, Logic Analyzer, Protocol decoder, Signal generator

Make it with Micromite

Phil Boyce – hands on with the mighty PIC-powered, BASIC microcontroller

Part 24: Counting pulses, rotary encoders and a digital safe

This month, we are going to explore three useful MMBASIC commands that relate to counting digital pulses. Using these built-in commands greatly simplifies tasks such as measuring the frequency of a signal, measuring the time period of a pulse cycle, or simply counting the number of pulses present on an input pin.

Linked to these topics is something that follows on nicely from last month, where we used a rotary potentiometer to generate a voltage between 0V and 3.3V, which in turn was used to control the position of a servo actuator arm. Several readers contacted us and asked if there was an alternative to a potentiometer; something that can instead be continually turned in either direction (unlike a potentiometer which is mechanically limited – typically to around 270°, not even a complete 360° revolution). This is where rotary-encoders come into play, and if you have never used one before, they can be an extremely useful input device. To make this a fun, we will show you how to use a rotary encoder to simulate entering a digital safe combination number.

What exactly are digital pulses?

Digital pulses comprise transitions between low and high logic levels on a signal line. The result is a digital signal. These can vary considerably, yet are often simply drawn as the examples shown in Fig.1. The pulses in a digital signal can be repetitive and symmetrical; an example is a square-wave – see Fig.1a. Here they are shown with a fixed frequency (ie, the length of a pulse cycle is constant), and with a duty cycle of 50% (meaning that for 50% of the pulse cycle, the logic level is high, and for the remaining time it is at a low logic level). We have seen, and used, square-waves earlier in the series when we used a PWM signal to drive a piezo sounder to generate musical notes. Pulses can also be repetitive but non-symmetrical ie, where the frequency is constant, but the duty cycle is not 50% – see Fig.1b. A great example of this is a servomotor signal (ie, PWM signal) where, as we saw last month, the duty cycle controls the position of a servo motor actuator arm. Digital pulses may also be random (ie, non-repetitive and non-symmetrical) – see Fig.1c where neither the frequency nor the duty cycle is constant.

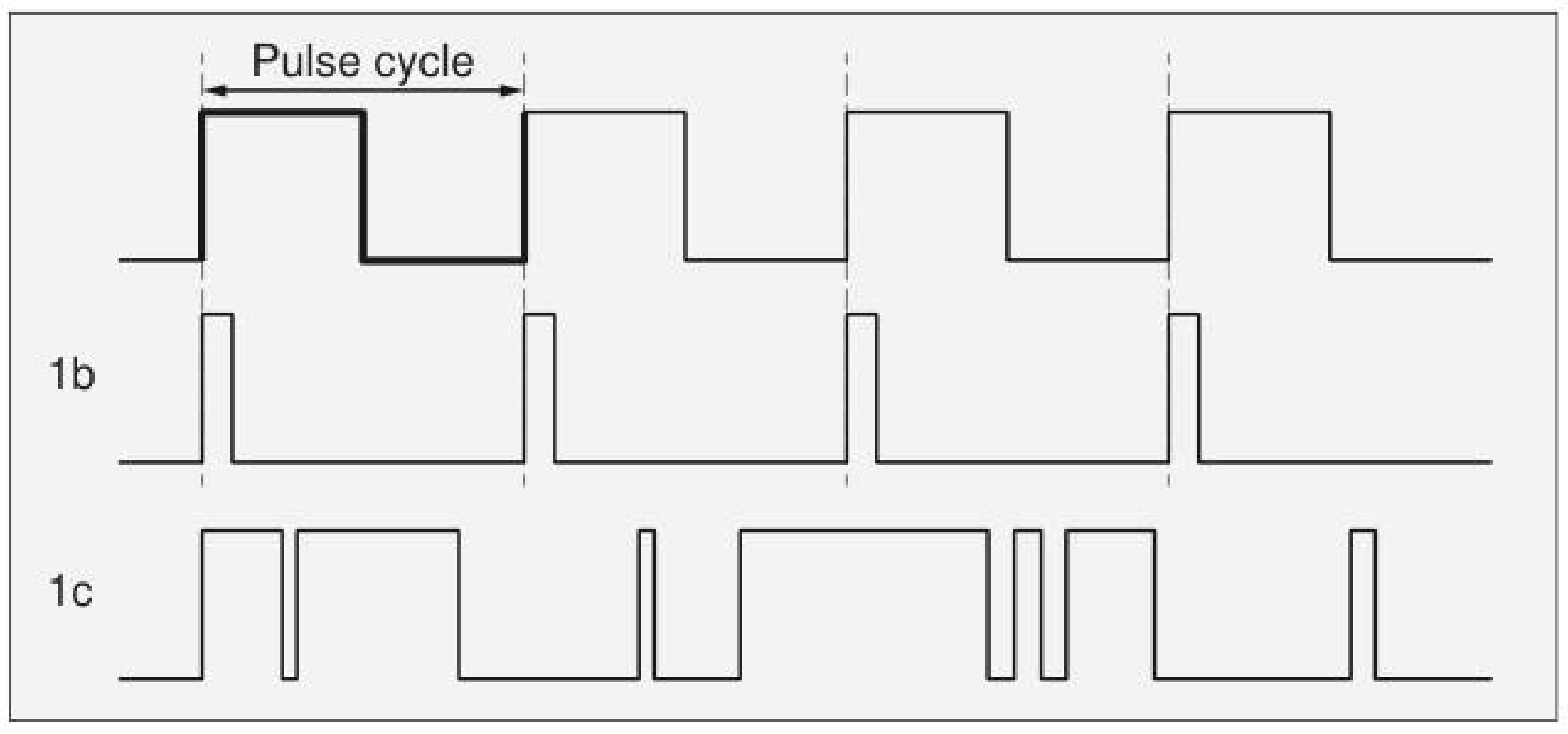


Fig.1. Examples of various digital signals. 1a represents a square wave, meaning a duty cycle of 50%. 1b represents a digital signal with the same frequency (ie, fixed time-cycle) as 1a, but with a lower duty-cycle. 1c is a random signal with a varying frequency (and cycle time period) and with varying duty-cycles.

Consider each of these digital signals being drawn as a graph where the *x*-axis represents time, and the *y*-axis represents the digital logic level; ie, either a logic high (3.3V) or logic low (0V). The point here is that at any moment in time, the state of the digital signal can be considered as either a low or a high logic state.

If we were to use a Micromite input pin to read the logic level of a digital signal, then by detecting the transitions between the low/high states within the signal, we can start to measure certain parameters about the digital signal. We'll discuss the theory first and then look at the MMBASIC commands that simplify the whole process.

Pulse cycle

A pulse cycle within a digital signal can be considered as a 'complete wave' that simply comprises a high-level logic pulse and also a low-level logic pulse (it doesn't matter which comes first). Fig.1a has one square-wave pulse cycle highlighted in bold to make this a little clearer. Note too that the boundaries of each pulse cycle are indicated by the dotted lines in Fig.1a and Fig.1b. In terms of logic-level transitions, a pulse cycle in a digital signal is shown in Fig.1 as starting with a low-to-high transition followed by a high-to-low, and finishes on the next low-to-high transition (which is effectively the start of the next pulse cycle, and so on…). Understanding this concept of a pulse cycle allows us to measure two important parameters – time period and frequency – by simply detecting logic-level transitions.

Pulse time-period

So, having just explained what a pulse cycle is in terms of logic-level transitions, it now makes it easy to explain the pulse time period (sometimes referred to as 'cycle time'). The time period is simply the time taken to complete one pulse cycle. In other words, relating to the square-wave pulse cycle highlighted in bold in Fig.1, it is the time taken between two consecutive low-to-high logic-level transitions. More on this later when we discuss the MMBASIC command to measure this timing.

Signal frequency

Frequency is often regarded as the number of pulses per second, and it is measured

Questions? Please email Phil at:
contactus@micromite.org

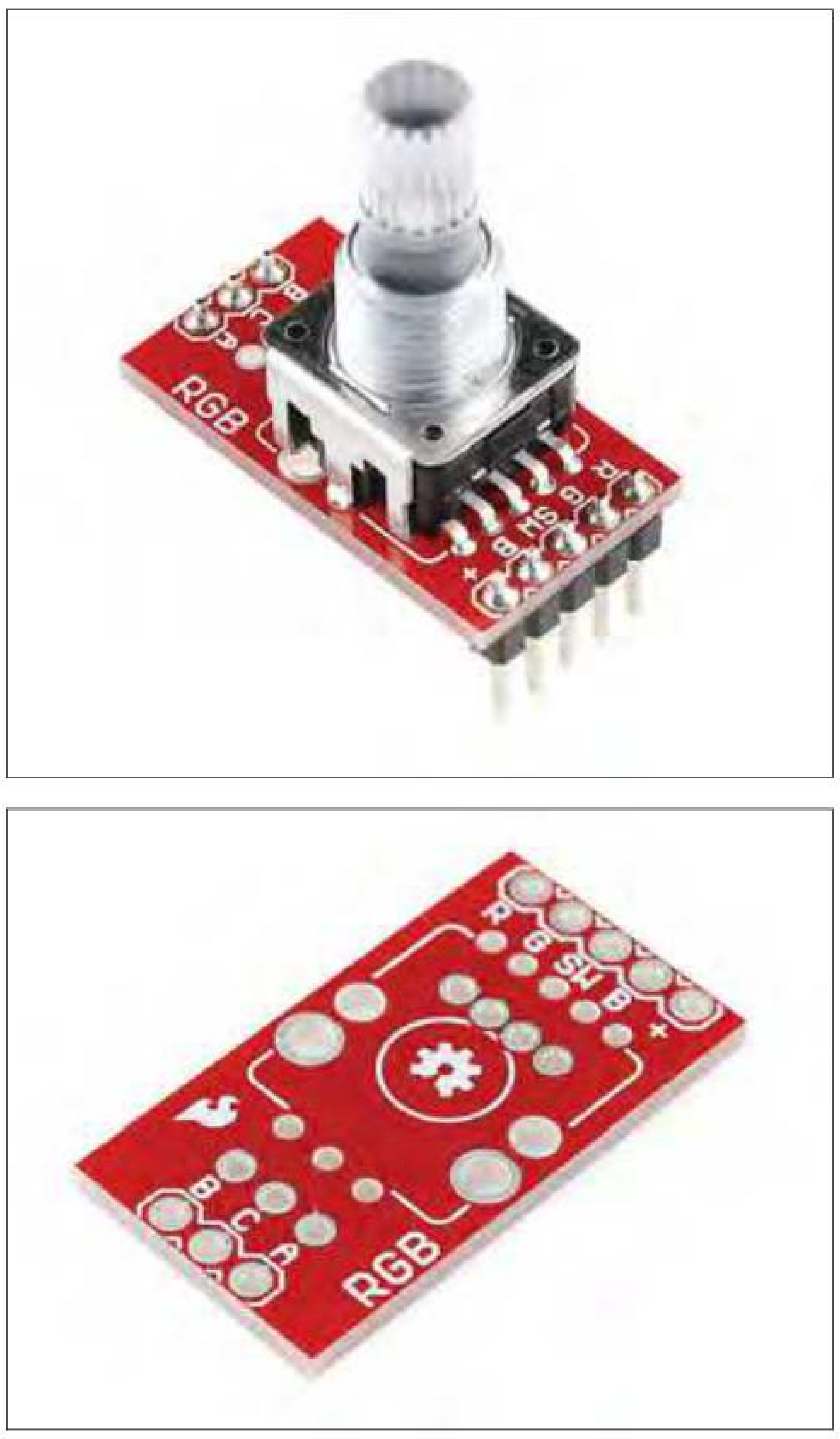


Fig.2. This rotary encoder comes with a built-in RGB LED and a push-button (activated by pressing on the shaft). A breakout board is available, making it a breadboard-friendly device.

in hertz (abbreviated to 'Hz'); but more technically correct, it is the number of *complete* pulse cycles per second. To determine the frequency of a digital signal, we could start a 'one-second timer' on a low-to-high transition, and then count the number of subsequent low-to-high transitions while the timer is 'active'. When the one-second interval is up, the count reached would represent the frequency (in hertz) of the digital signal.

An alternative is to use the well-known equation: *Freq* = 1/(*time period*)

If we measured a time of 0.1 seconds for a single pulse cycle, then the measured frequency would be 1/0.1 = 10Hz. Note that by using the equation method, we do not have to wait for the 'one-second' timer duration.

The theory described above shows that for MMBASIC to measure a time period or the frequency of a digital signal, all it needs to do is detect logic-level transitions and accurately measure the time between such transitions.

We could write a BASIC program to do this, but what if we want to measure a high frequency signal – there would come a point where our program code would simply be too slow to keep up. To overcome such a limitation, MMBASIC has three commands built into the firmware that automatically detect these logic-level transitions at an extremely fast rate.

The `SETPIN` command

As we have seen earlier in this series, the `SETPIN` command is used to configure a Micromite pin to behave in a certain way. The simplest syntax of this command is: `SETPIN` *`pin-number, configuration, option`* where:

`pin-number` is the value of the physical pin number on the Micromite chip. It must be a valid value otherwise an error is generated (see the *Micromite User Manual* for specified values).

`configuration` is a parameter that determines the function of the specified pin. When we first introduced the `SETPIN` command early on in the series we used it to define a pin as a digital input with `SETPIN x,DIN` (where *x* is a valid pin number) and used `SETPIN x,DOUT` to define a digital output. Last month, we also used `SETPIN x,AIN` to define an analogue input pin to which we connected a potentiometer, which supplied an analogue voltage between 0V and 3.3V.

`option` is an optional parameter that may define further details; for example, previously we have used `PULLUP` or `PULLDOWN` in conjunction with `DIN` when using a push-button as an input. This avoided having to use a physical resistor to tie the input signal to a default logic level (with a push of the button then setting the opposite logic level).

So, why have we explained all this? The answer is that the pulse-counting commands that we are exploring this month are also implemented with the `SETPIN` command, but with different *`configuration`* parameters.

The `FIN`, `PIN`, and `CIN` parameters

The `SETPIN` command *`pin-number`* parameter must be a valid value, and referring to the *Micromite User Manual*, it explains that four pins (pins 15-18) are valid for use as 'COUNTing' inputs on the 28-pin Micromite. Note that 'COUNTing' pins means that `FIN`, `PIN` and `CIN` can be used on any of these pins (and not just the counting (`CIN`) functionality). Now let's demonstrate each in turn, starting with how the *MKC* can measure the frequency of a digital signal.

Note that the voltage levels of any digital signal that we wish to measure must not exceed 5V, and ideally be at 3.3V for the *MKC*. If you wish to measure a digital signal that has a higher voltage, then simply add a potential divider between the high-voltage signal and 0V. For our demonstration purposes, we will simply use a Micromite output pin as the source of the digital signal, and so it will not exceed 3.3V for a high logic level.

Using `FIN`

To demonstrate how to measure the frequency of a digital signal, we will use pin 16 as the input pin. We could also use pin 15, 17 or 18, but pin 15 may already be in use if you have a touch-screen connected and enabled. For a digital test signal, we will use the `PWM` command to output a square wave on pin 4. There is no diagram for this circuit, since all we need to do is connect a jumper wire between pin 4 and pin 16. Once you have done this, connect your *MKC* to your computer, start the Terminal app, and enter the following:

```
Freq=2000
SETPIN 16, FIN
PWM 1, Freq, 50
DO
  PRINT PIN(16)
LOOP
```

Before you `RUN` the code, let's explain what we're doing. The first line sets a variable named `Freq` with a value of `2000`. This will be the frequency (in hertz) of the generated square wave from the `PWM` pin (ie, the frequency of the digital test signal).

The second line configures the appropriate input pin (pin 16) to which we are connecting the data signal that is having its frequency measured. The third line starts to output a square wave on pin 4 (`PWM channel 1A`). It is a 'perfect' square wave with a 50% duty-cycle; and the frequency is set to the value stored in the `Freq` variable. Next, a `DO…LOOP`, in which we continually `PRINT` the value of `PIN(16)` on the terminal screen. Pin 16 is configured with the FIN parameter, so when you `PRINT PIN(16)` you don't get the logic value (as you would if we configured pin 16 with `DIN`), but instead the firmware measures the frequency by counting the number of low-to-high transitions that occur over one second.

`RUN` the program and you should see is a stream of '0's (zeros) initially displayed on the screen, followed possibly by a stream of values close to `2000`, and finally after one second it should settle on a stream of `2000`. If you do not see this, check your code matches that shown above, and also check you have pin 4 and pin 16 connected to each other. Note that the `2000` shown is a value in hertz; ie is equal to 2kHz.

Gate time

The reason for the initial zeros (and potentially a few non-2000 values) is that there is a one-second period of time during which the firmware is busy counting the low-to-high transitions in the background. This period of 'counting time' is referred to as the 'gate time'; it must elapse before

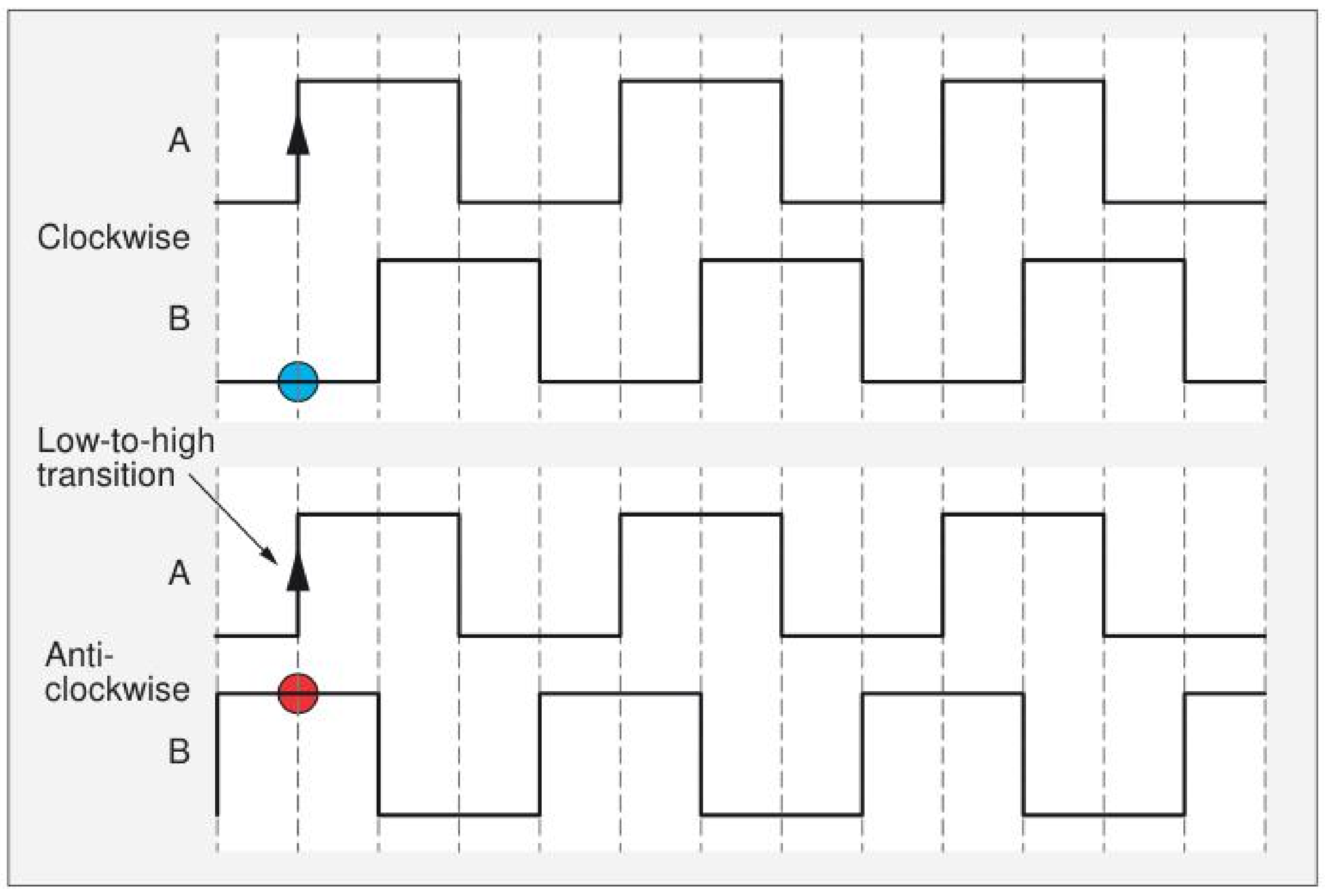


Fig.3. The rotary encoder outputs two digital signals: ‘A’ and ‘B’. By detecting a low-to-high transition on ‘A’, we can read the logic level of ‘B’ to determine the direction of rotation.

a frequency count can be provided. Gate-time, therefore, determines how often the output result is updated (important if the frequency of the data signal varies).

Now refer back to the syntax of the `SETPIN` command and you’ll see there is an *`option`* parameter. An important point here is that when you use the `FIN` *`configuration`* parameter, the *`option`* parameter defines the gate time in milliseconds (ms). So why would we want to adjust the gate time? Well, if the frequency of the data signal is very high, then you could count many pulses in a shorter space of time and achieve an accurate result quicker. In this scenario, the output would be updated more often (shorter gate times mean the count result is displayed sooner). Similarly, if the data signal has a very low frequency, then a much longer gate time is required in order to count a number of pulse cycles in order to measure the frequency. You couldn’t use a gate time of one second to measure a signal that only changed every 10 seconds!

To see the above in practice, stop the program (`Ctrl-C`), and then `EDIT` the program to set the Freq variable to `200000` (ie, 200kHz) – then `RUN` the program again. We are still using a one second gate time, so there is still an observable ‘delay’ before the result is shown.

Next, shorten the gate time to 10ms by changing the second line to `SETPIN 16, FIN, 10` and on running the program you should see the resultant frequency displayed much quicker (every 10ms).

In summary, when using the `FIN` configuration parameter, ensure you use an appropriate gate time to allow MMBASIC to count a decent quantity of pulse cycles (which is ultimately determined by the frequency of the input data signal). The gate time parameter value must be between 10 and 100,000; and the outputted value is always in hertz, regardless of the gate-time value.

When measuring signals with a frequency less than 10Hz, it is often better to measure the pulse time period and then use the formula *Freq* = 1/(*Time Period*). This is what we will examine next by using the `PIN` *`configuration`* parameter instead.

Using `PIN`

To measure the cycle time period, the Micromite firmware just needs to accurately measure the time between two successive low-to-high transitions. By using `SETPIN 16, PIN` in the above program, we can see this time period measurement in action.

Change the second line (with no *`option`* parameter), and also set the value of `Freq` to `50` (ie, 50Hz). On running the program, you will see the value `20` displayed, which means the period of the 50Hz signal is 20ms. The result is always in ms – a value of 20 means 20ms (0.02s). As a sense check, note that *Freq* = 1/(*Time-Period*) = 1/0.02 = 50Hz.

When the *`configuration`* parameter `PIN` is used with the `SETPIN` command, the *`option`* parameter can be used to determine over how many pulse cycles the measurement is taken to average the period measurement. Valid values are 1 (default) to 10000. You would tend to set this to a higher value when measuring the time period of higher frequency signals.

To see this in action, set the `Freq` value to `5000` (ie, 5kHz), and run the program. The resultant time period displayed is not really accurate, so now add the *`option`* parameter with a value of 1000 so that 1000 pulse cycles are timed and averaged. On running the program, you will now see the value 0.2 displayed. This represents a time period of 0.2ms = 200µS, which is the period of a 5kHz signal.

In summary, when measuring high-frequencies, it is better to use `FIN`, and when measuring low-frequencies, it is better to use `PIN`.

Using `CIN`

The *`configuration`* parameter `CIN` is used if you simply just want to count pulses (or to be precise, count low-to-high transitions). On configuring a pin with `CIN`, an internal counter is reset to a value of zero. Then, on every low-to-high transition, the counter increments by one. Then, whenever you use `PRINT PIN(`*`x`*`)` (where *x* is one of the four valid `COUNT` pins), the value of the counter will be displayed. To reset the counter back to zero, simply use `SETPIN `*`x`*`,CIN` again. Note that the Micromite can detect input pulses as brief as 10ns, and hence if `CIN` is used, for example, to try and count button presses on a push-button, then any mechanical bounce as the button is pushed will also be counted (which is why it’s important to debounce mechanical switches). Therefore, `CIN` is much more suited to counting pulses in a true digital signal as we will now explore. Note, there is no *`option`* parameter when using `CIN`.

Using our existing code, set the `Freq` value to `1000`, ie 1000 pulses will be generated every second. Then change the second line to `SETPIN 16, CIN` and run the program. You will see the pulses being counted, with effectively the number of ‘thousands’ being the number of seconds the program has been running for.

Have a play by changing the value of `Freq`, and see how this affects the displayed value after each second.

The above shows how to use the Micromite to perform various counting tasks on a digital signal. We will now look at a slight variation to this in the form of detecting pulses from a rotary encoder.

Rotary encoders

A rotary encoder is a rotary position sensor, but it can also be used as a ‘digital potentiometer’ which has no limitation on its rotation – it can be continually rotated in either direction. Instead of containing a variable resistance, a rotary encoder simply outputs a sequence of digital pulses, often as two separate digital outputs. By knowing how to interpret these pulses, we are able to determine the direction (and speed) of rotation – more on this shortly.

Rotary encoders come in many different styles; however, we really like the one shown in Fig.2. It has a built-in RGB LED which can be made to illuminate through the clear plastic shaft; and by pushing down on the shaft, you can activate the built in push-button. In addition, while rotating the shaft, it has a soft ‘click action’ which provides nice tactile feedback. These features are not all normally found on a rotary encoder; and when you consider that it is readily available online at low cost, you can see why it is our favourite rotary encoder. To make it breadboard friendly, an optional breakout-board is also available (see Fig.2).

Decoding the pulses

The pulses generated by this rotary encoder are output on two digital signal lines – labelled ‘A’ and ‘B’ on the breakout board. These pulses are actually generating a 2-bit Grey-code signal. We won’t explain the details of Grey code here, but Fig.3 shows the kind of signals it creates. The point to observe here is that there are two possible rotations – you can turn it clockwise, or anti-clockwise. So how do we decode these pulses? The answer is actually quite simple once you apply the

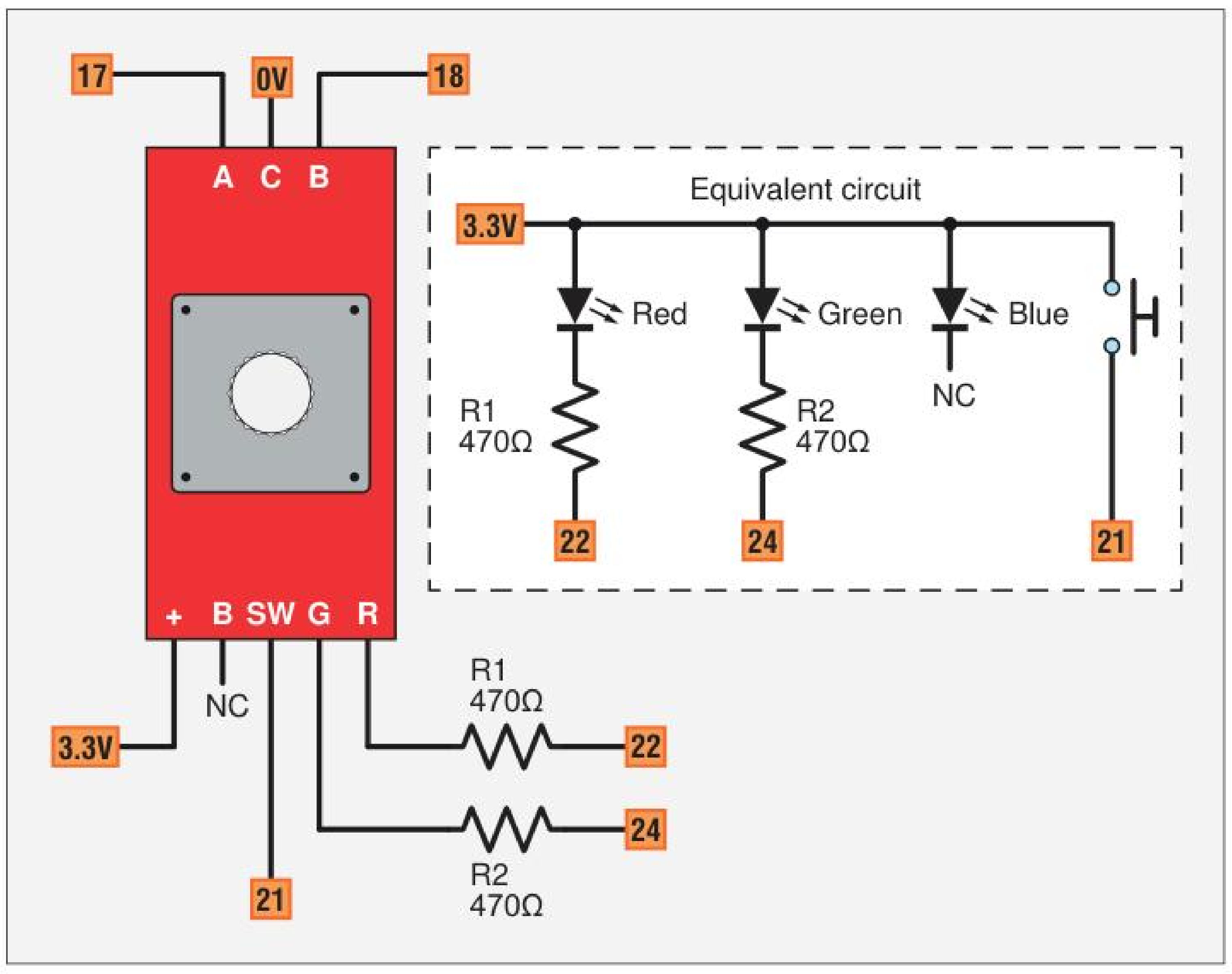


Fig.4. The rotary encoder connects to the *MKC* with three connections to read the output pulses. The *Digital Safe* project adds connections for the RGB LED and the push-button.

following technique. Referring to Fig.3, consider the moment in time when there is a low-to-high transition in signal A. If signal B is at a low logic level (highlighted by the blue circle), then the shaft is rotating clockwise. However, if signal B is at a high logic level (highlighted by the red circle), then the shaft is rotating in an anti-clockwise direction. It's that simple!

Note that you could detect the transition of signal B instead, and then read the logic level of signal A to determine the direction. And you can also use a high-to-low transition – whatever option you select, just stick with it.

Let's put this into practice with a few lines of code. First, refer to Fig.4 for the circuit diagram. Note at this stage we only need to connect the three contacts labelled 'A', 'B', and 'C' to the *MKC* (respectively to pins 17,18 and 19). A and B are the two signal lines, and C is effectively a common that is connected to 0V. Connect this part of the circuit, and then enter the following short program:

```
SETPIN 17, INTH, MyInt, PULLUP
SETPIN 18, DIN, PULLUP
Rot_Value=50 : PRINT Rot_Value

DO : PAUSE 1 : LOOP

SUB MyInt
  PAUSE 0.1
  IF PIN(18)=0 THEN
    Rot_Value = Rot_Value + 1
  ELSE
    Rot_Value = Rot_Value – 1
  END IF
  PRINT Rot_VALUE
END SUB
```

The first line of code simply configures pin 17 to trigger an interrupt subroutine (called `MyInt`) whenever a low-to-high transition is detected on signal 'A'. The `PULLUP` option is used to tie the default logic-level high (with the rotary encoder simply pulling it low whenever it needs to output a low pulse). The second line configures pin 18 as an input so that the logic level on signal 'B' can be read when required. The third line sets a variable that we have named `Rot_Value` with a value of `50` – it also prints this value on the terminal screen.

A simple `DO...LOOP` is included, and acts as the main program. It does not do anything here, but is required so that the program can continue to run while waiting for a low-to-high transition on pin 17 (ie, from the rotary encoder's signal A output.

The interrupt subroutine is the part that then determines the direction of travel by reading the state of pin 18. Depending on the state of pin 18, `Rot_Value` is either incremented by 1 (clockwise), or decremented by 1 (anticlockwise). And before the subroutine is exited, the new value of `Rot_Value` is displayed on the screen. Note that the `PAUSE` command adds a small delay to remove any mechanical contact bounce generated by the rotary encoder.

`RUN` the program to ensure it works correctly by spinning the rotary encoder. Upon each 'click', you should see the value displayed on the screen increase or decrease according to the direction of rotation.

If you see the above work, but 'in reverse' (ie, increase with an anti-clockwise rotation), then you can either make a hardware change (by swapping the connections to pins 17 and 18), or you can change the code inside the interrupt subroutine – I will leave it to you to work out what to change!

Digital safe

Having seen how to use a rotary encoder to increase and/or decrease the value of a variable depending on which direction it is 'spun', it is now time to have some fun by demonstrating how a rotary encoder can be put to good use in a practical project – we will simulate entering a safe's combination-lock number in a short demo program. This will require the use of the red and green LEDs built into the rotary encoder, as well as the built-in push-button.

Now connect the rest of the circuit shown in Fig.4 (ie, connect the *MKC* to the push-button SW contact, and LEDs R and G via 470Ω resistors). Next, download and `RUN` the **DigitalSafe.txt** program (available from the January 2021 page of the *PE* website). On running the program, you will see the current 'combination dial' value displayed on the terminal screen. Simply turn the rotary encoder in the appropriate direction to enter the three required values of the combination: 3, 46, 22. When you reach the first required value, push the encoder shaft to activate the push-button, which in turn 'submits' the number displayed on the terminal screen. Continue with the second and third required values above in a similar manner. If you entered the three-number combination correctly, then the green LED will illuminate, otherwise the red LED will illuminate, meaning you failed to open the safe!

The code is commented throughout so we won't go into the details here, but do take the time to look through the code to see how easy it is to create this 'digital safe'.

Challenges

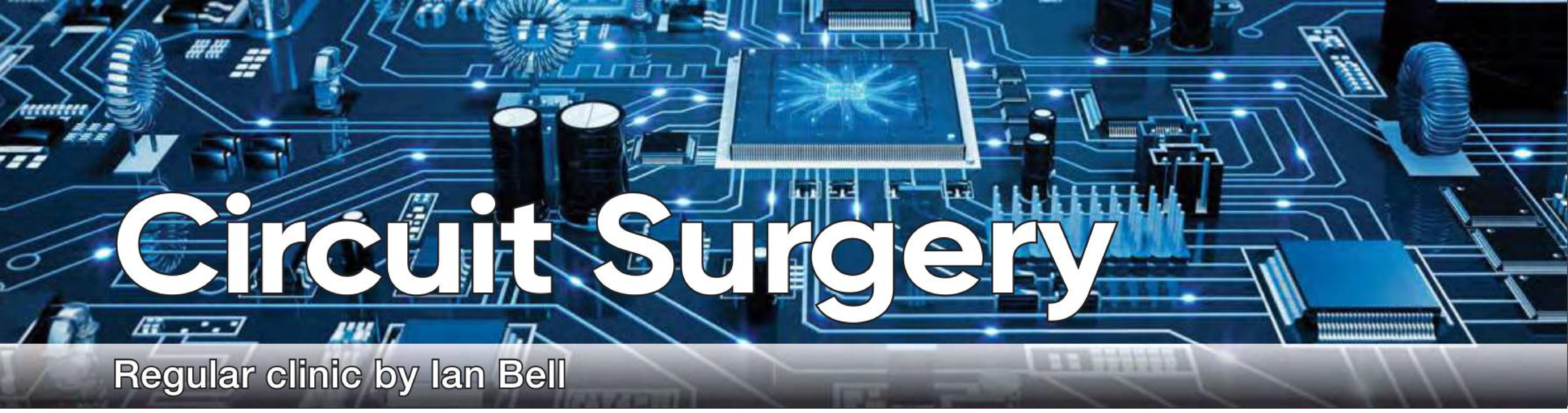
Once again, there are many things you can change in the program code to affect the way it works. In this example, why not add hardware too – here are some challenges:

1. Increase the range of each value from between 1-50 to between 1-99 (or greater)
2. Make the safe more secure by increasing the length of the combination number from three values to five values (or more)
3. Add a time-out feature; ie, the combination has to be entered within a certain amount of time
4. Add a piezo sounder to give audible feedback to signify a correct combination entry, and a different sound if incorrect
5. Add a relay and a solenoid to complete the lock!

Next month

We have covered a wide range of topics throughout the series, and along the way encouraged you to experiment and build your own Micromite-based projects. Several of you have written in to ask if there are more powerful versions of the Micromite. For example, devices that can run faster, or have more memory, or have additional features and commands for controlling different hardware, such as SD cards for storing data, or for driving larger displays with higher resolutions. The answer to all of these is very much *yes*, there are indeed different Micromites available, as well as different versions of MMBASIC. Next month, we will provide a summary overview of what else is available for those of you wanting to advance beyond the power of the *Micromite Keyring Computer*.

Until then, have fun coding!



Circuit Surgery

Regular clinic by Ian Bell

Interference and noise

Recently, Michael Lamontagne posted a question on the EEWeb forum concerning a possible ground loop problem when measuring an op amp circuit with an oscilloscope. Rather than looking at this specific case in detail (partly because we are not certain enough about exactly how everything was wired up) we will take a quick look at unwanted signals in general, specifically noise and interference.

Noise and interference

Unwanted signals present in electronic systems get often referred to as 'noise', although we can be more precise with our terminology. In audio system it may make its present felt as hiss, hum, buzzes and crackles. In sensor systems, it limits measurement of low-level signals and degrades accuracy of measurement. Noise may already be present as part of an input signal (eg, it may come from a sensor along with the wanted sensor signal), or it may be introduced by the circuitry (eg, amplifier) used to process the signal. For example, all resistors generate random electrical noise – you cannot prevent this, it is part of their basic physics. Circuits also produce non-random unwanted changes to signals – distortion, which we will not be looking at in this article. Unwanted signals may also come from outside or elsewhere in the system, coupled or picked up inadvertently and added to the signal being processed – this is often called 'interference' to distinguish it from random noise. Crosstalk is interference between multiple channels or signal paths.

Random noise

Radom noise causes the instantaneous value of a signal to deviate from its 'true' value, with decreasing probability for larger deviations. The specific mathematical probability function depends on the type of noise, but may be the well-known Gaussian or normal distribution, familiar to all statisticians. There are a variety of types of random noise generated within electronic circuitry; these include thermal noise, shot noise, flicker noise, and avalanche noise. This noise generated is fundamentally due to the discrete nature of electricity at the atomic level – electric charge in circuits is carried in packets of fixed size (eg, electrons).

The waveform in Fig.1 shows a random voltage variation with time. This gives us some simple insight into what noise 'looks' like, but in general, plotting random noisy signals against time is not particularly useful. When dealing with noise we often need to look at the spectrum of the signal – the variation of signal level against frequency.

Unwanted signals may look like random noise (eg, on an oscilloscope), but they can actually have significantly different characteristics. For example, the noise on the power supply of a digital circuit may look random, but a look at the spectrum will show that certain frequencies, related to the system clocks will be dominant. Pure random noise has a smooth continuous spectrum – for example, that shown in Fig.2.

Random noise may be classed according to the shape of its spectrum. White noise has the same power throughout the frequency (*f*) spectrum, whereas 1/*f* noise (or pink noise) decreases in proportion to frequency. For 1/*f* noise there is the same amount of noise power in the bandwidth of say 100Hz to 1kHz, as there is in 1kHz to 10kHz, whereas for white noise there would be 10 times as much power in the bandwidth 1kHz to 10kHz as 100 to 1kHz because it is 10-times larger. Amplifiers (and other circuits) typically exhibit a mixture of pink and white noise, with pink noise dominating at low frequencies. The frequency at which the dominant noise component changes between pink and white noise is called the corner frequency or noise corner (see Fig.2).

The fact that the components in any electronic circuit or system generate random noise means that there is always a certain level of noise, even with no signal present. This is known as the noise floor, which is important because the circuit cannot meaningfully process input signals that are smaller than the noise floor. As noise floor relates to noise within the circuit, this is different from noise within the input signal. If the properties of the required signal are known then there are techniques which can extract signals that are smaller than noise present within the signal.

The difference between the signal and the noise is often important; it is expressed as the 'signal to noise ratio' (SNR), usually in decibels (dB) and based on the ratio of noise power (hence the v^2 terms in the equation). *Larger* values indicate *better* performance.

$$SNR_{dB} = 10\,log_{10}\left(\frac{v_s^2}{v_n^2}\right) = 20\,log_{10}\left(\frac{v_s}{v_n}\right)$$

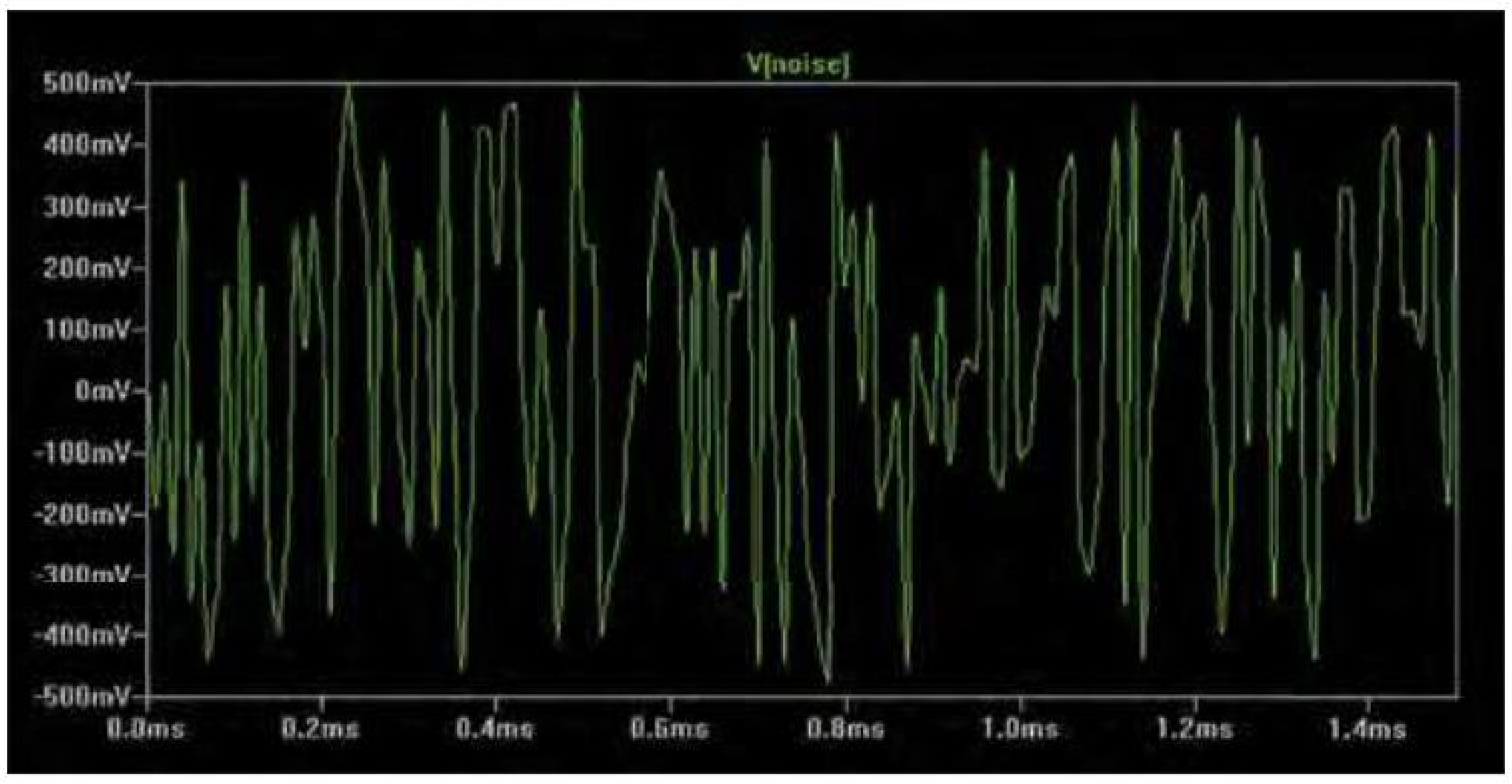


Fig.1. Random signal (noise).

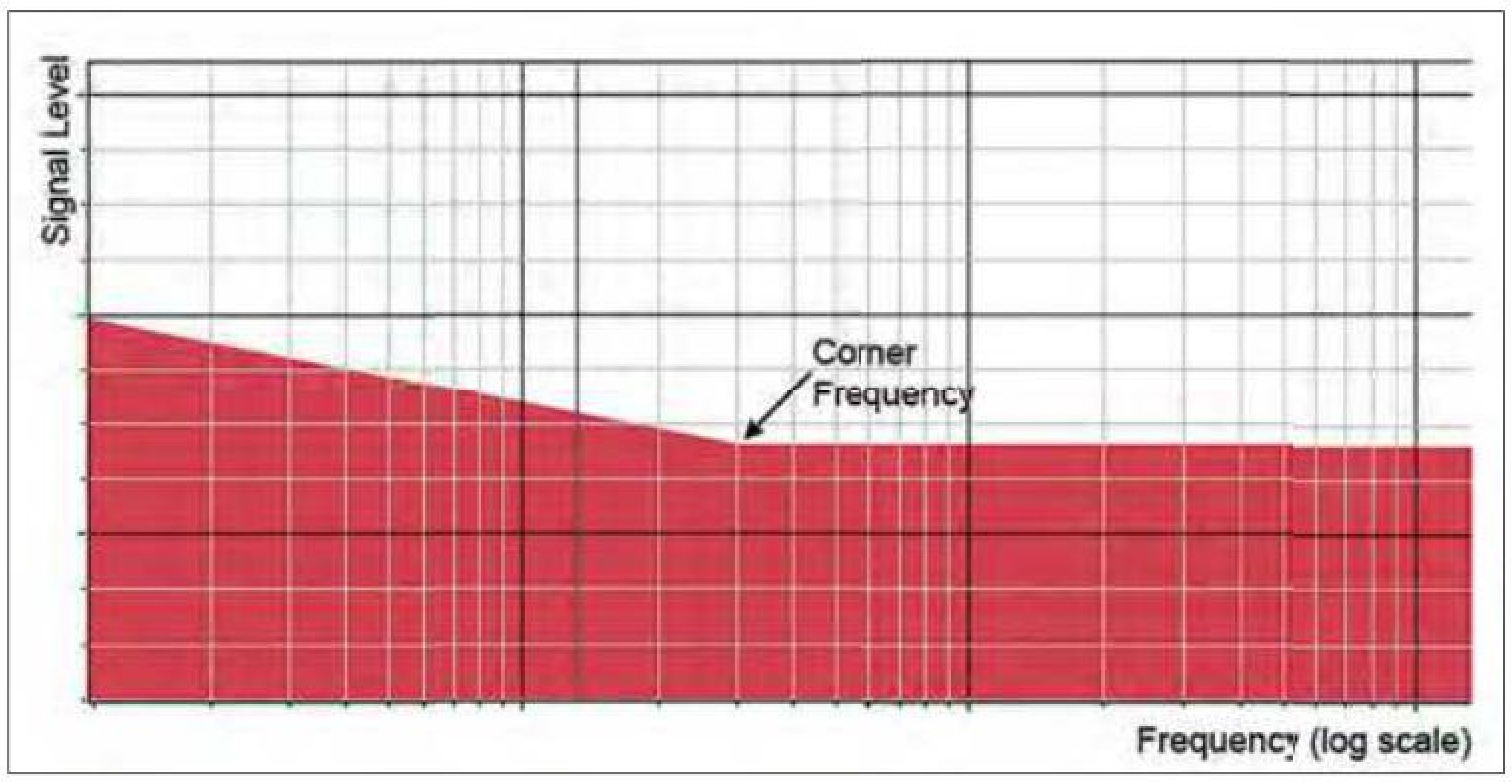


Fig.2. Typical spectrum of amplifier noise.

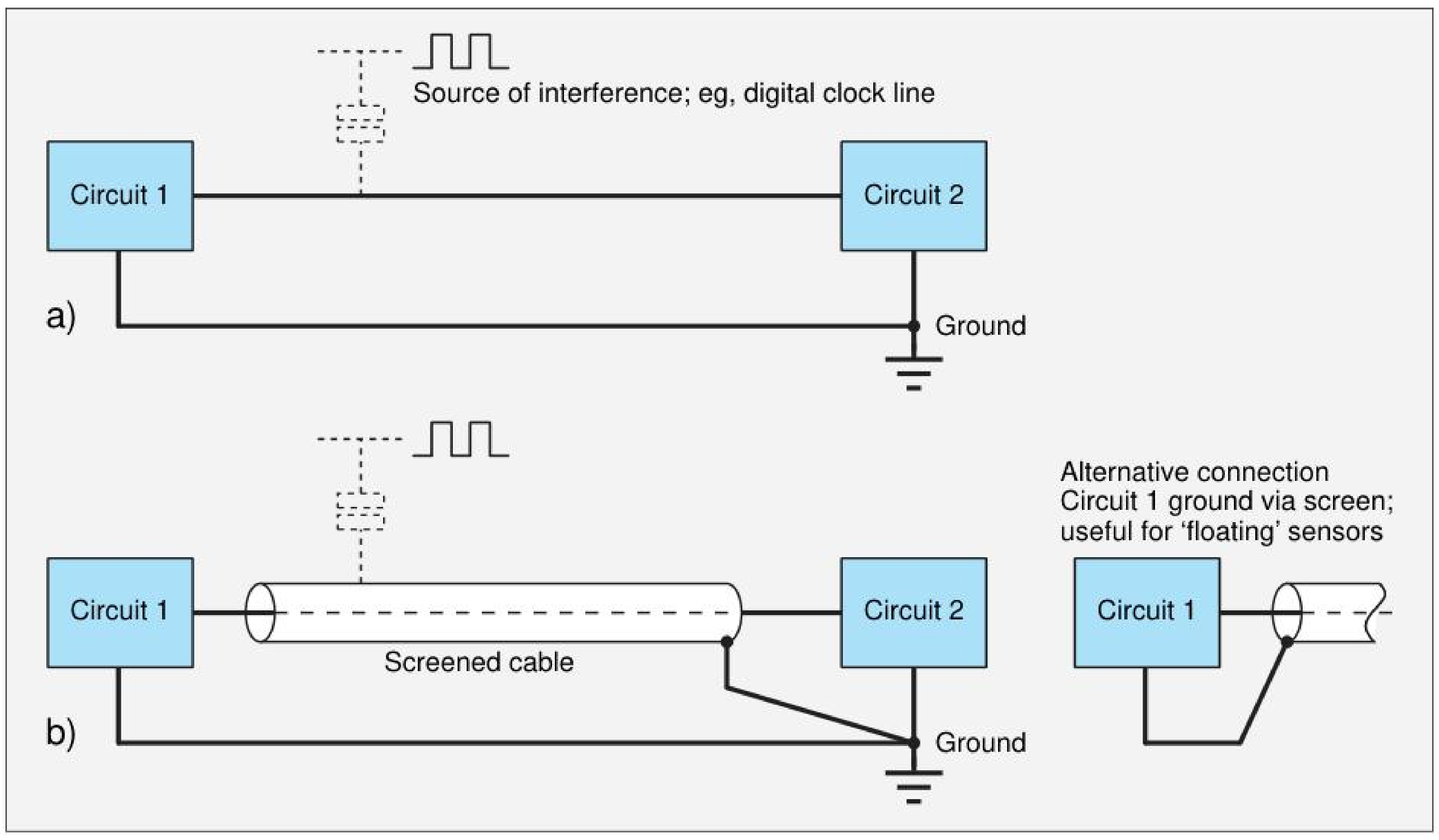

Source of interference; eg, digital clock line
Circuit 1
Circuit 2
a)
Ground
Alternative connection Circuit 1 ground via screen; useful for 'floating' sensors
Circuit 1
Circuit 2
Circuit 1
Screened cable
b)
Ground

Fig.3. Capacitively coupled interference.

Here, v_s is the rms signal voltage and v_n is the rms noise voltage. When using or quoting SNR values, the bandwidth (range of signal and noise frequencies considered) should be quoted because noise power is frequency dependent and noise may be present well outside the range of signal frequencies of interest.

Interference

External signals may get into your circuit through electrostatic, electromagnetic and magnetic coupling. In electrostatic coupling, a high-impedance part of your circuit acts like one plate of a capacitor; in magnetic coupling, a loop in your circuit acts like the secondary of a transformer; and in electromagnetic coupling, parts of your circuit act like antennas. Mains hum signals (at 50/60Hz) and radio frequency interference from other electronic systems such as phones and computers are obvious examples of external interference. The amount of external noise a circuit is subject to will vary greatly depending on its location. The problem will tend to be worse close to things like power lines, electrical machines and transmitters such as mobile phones.

Signals in one part of your circuit can find their way into other parts of the circuit where they cause problems. A common example of this is the clock of a digital section of a mixed analogue and digital circuit getting into an analogue section, via the power supply lines or by capacitive coupling to high-impedance points. Careful circuit design and construction can greatly reduce these problems.

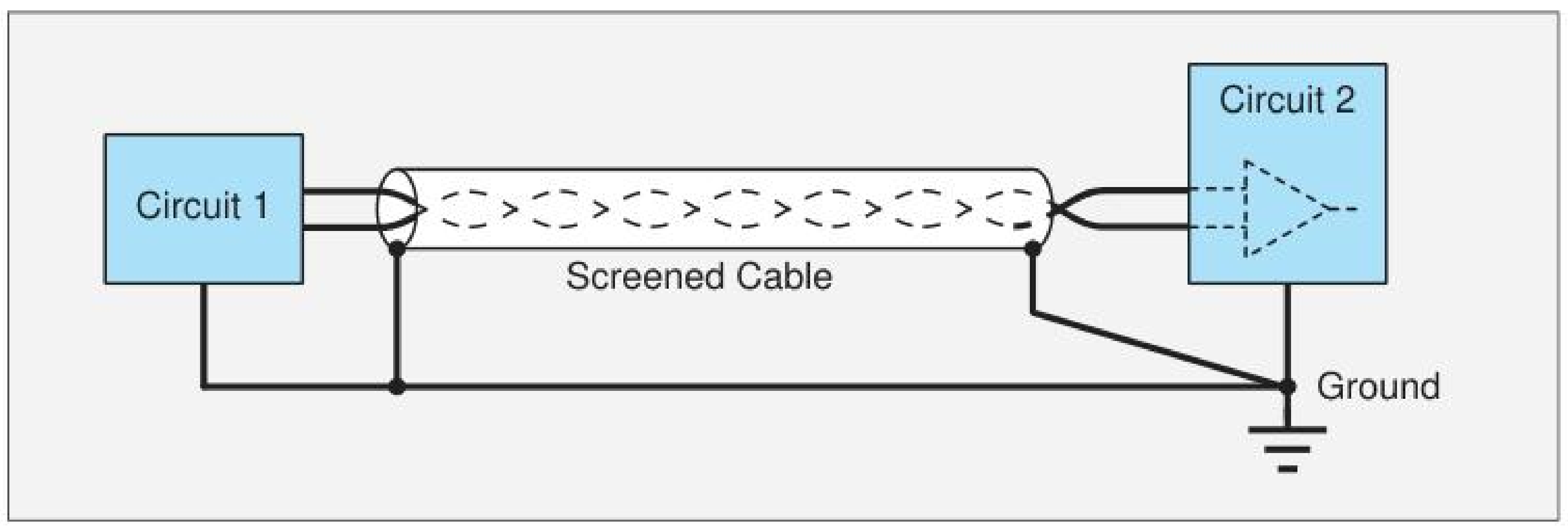


Fig.4. Screened differential signal.

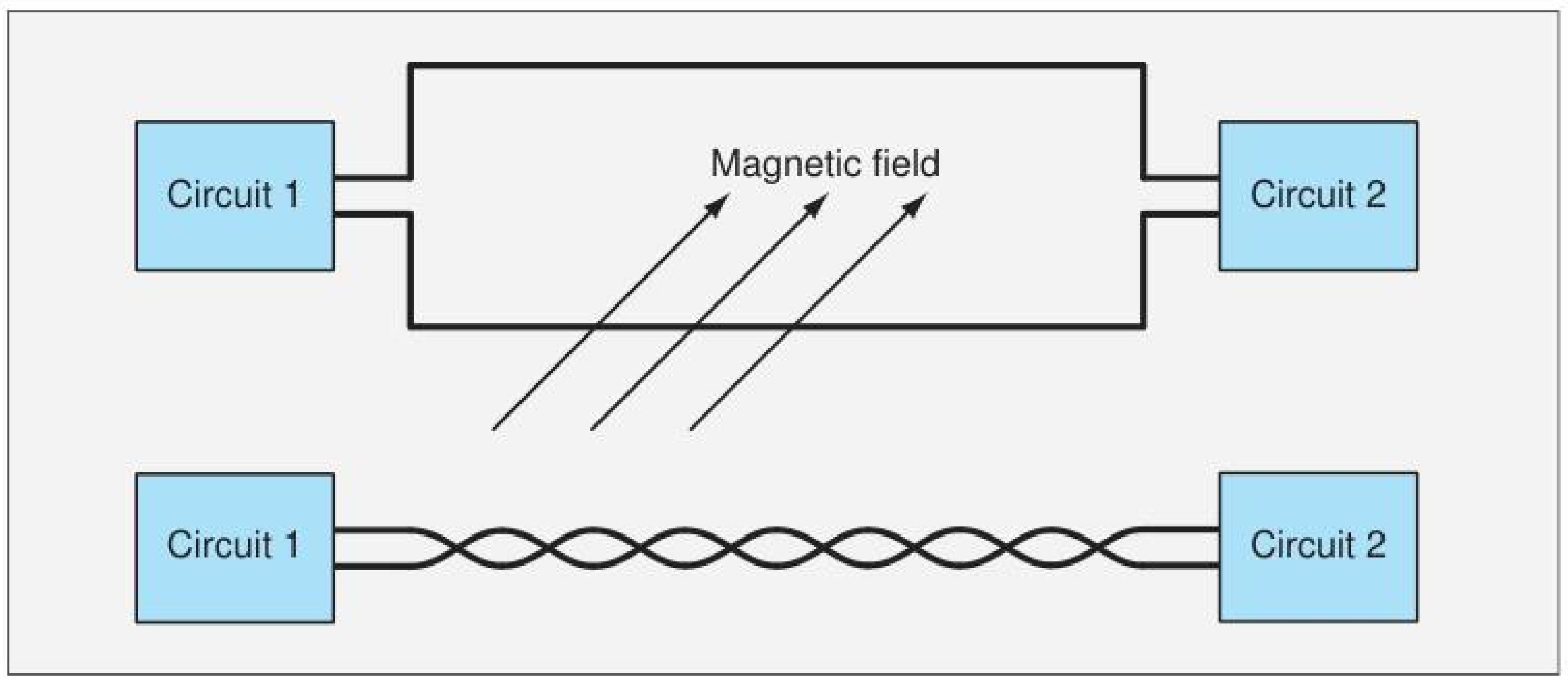


Fig.5. Large wiring loops (upper schematic) make a circuit susceptible to voltages generated by magnetic fields. Reducing loop size (lower schematic) helps combat the problem.

Capacitively coupled interference (see Fig.3) can be reduced using screening, which effectively grounds the interference coupling capacitance. Screening is implemented using coaxial (screened) cable to link (for example) a sensor to a circuit, and by enclosing the sensitive circuits in a grounded screening box. The source of interference can also be screened to reduce its effect on other circuits. Choice of where the screened cable is grounded may have an effect on circuit performance due to the possibility of creating grounding loops if the screen/signal return path is grounded at both ends (more on this a little later). For differential signals, we can also use screened wires – the two signal wires form a twisted pair and are enclosed by the screen (see Fig.4). Here, grounding at both ends is less of a potential problem as the ground does not carry the signal.

Magnetic interference is worse when physically large loops occur in the circuit (see Fig.5) but can be reduced by avoiding such loops – for cables, use of twisted pairs of wires is an effective approach. For PCBs, use a 'ground plane' on one side of the board and for ribbon cables each signal can be given an adjacent ground wire. Circuits can be shielded against magnetic fields, but this is not used as commonly as shielding for capacitive coupling as it requires special high-permeability materials such as Mu-Metal. These materials are expensive and since they may need to be quite thick the screening may be bulky.

Power supplies can be a significant source of unwanted signals. The circuit in Fig.6 illustrates how supply resistance leads to errors or interference. The supply current taken by a circuit causes a voltage drop across the supply wiring; so, for example, the 'ground' voltage at each subcircuit will not actually be the same (as measured from the same reference point). Fig.6 is a simplification – every branch of the supply wiring will have resistance and hence cause voltage drops.

Supply and group voltage drops may cause problems if we are trying to accurately process voltage signals. If the supply currents are constant, then the error will be an offset (DC error) but the ground voltage is not necessarily constant; as the supply current of one subcircuit varies then the supply voltage drop and hence the error voltage at this and other subcircuits fluctuates (this is sometimes called 'ground bounce'). This problem can be very significant, for example, when one subcircuit has a digital clock signal that is coupled via the supply into a sensitive amplifier. This situation can also occur in the mains wiring – along the equipment power cables and the wiring between supply outlets.

A solution to supply and ground noise is to wire connections separately to a single point rather than using the same point-to-point wire for all the connections (see Fig.7). This approach applies equally to the wiring inside the cabinet of an instrument and to the supply connections on an integrated circuit. It may be more difficult to achieve with mains wiring, as it cannot be easily changed and doing so may be very dangerous.

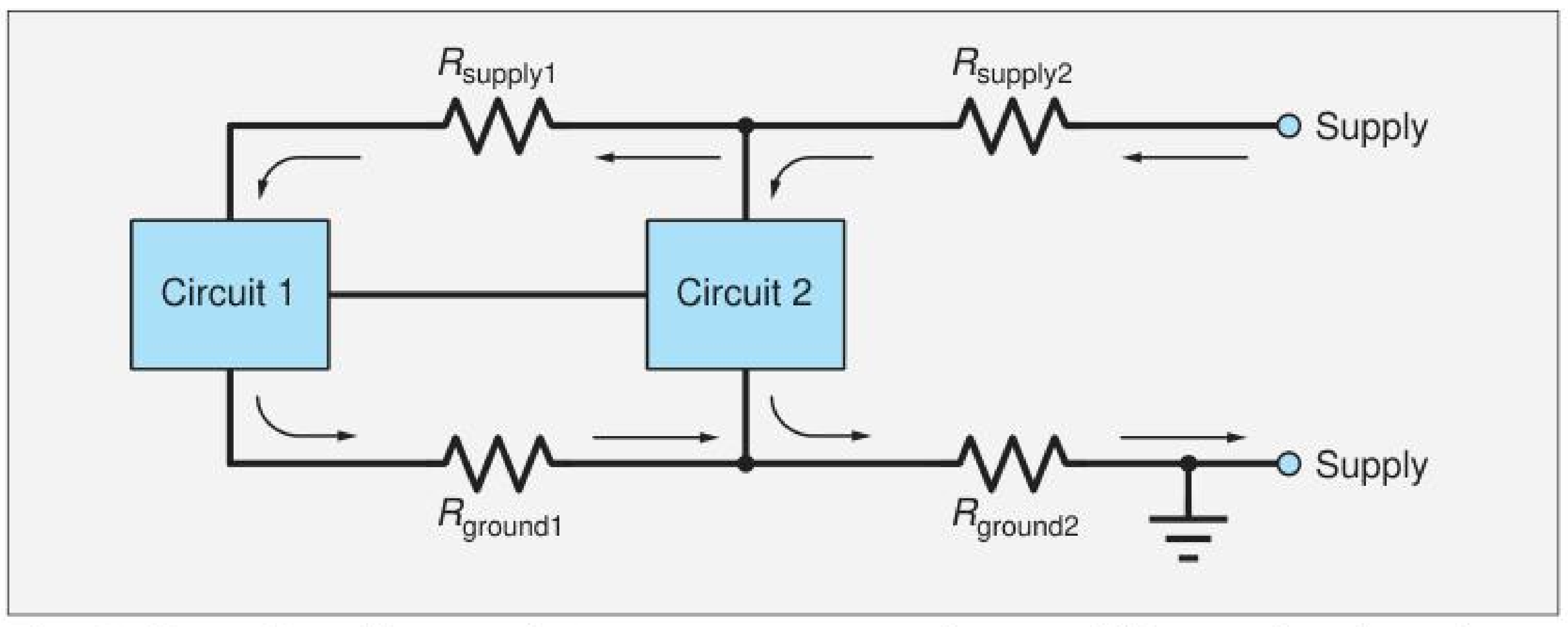
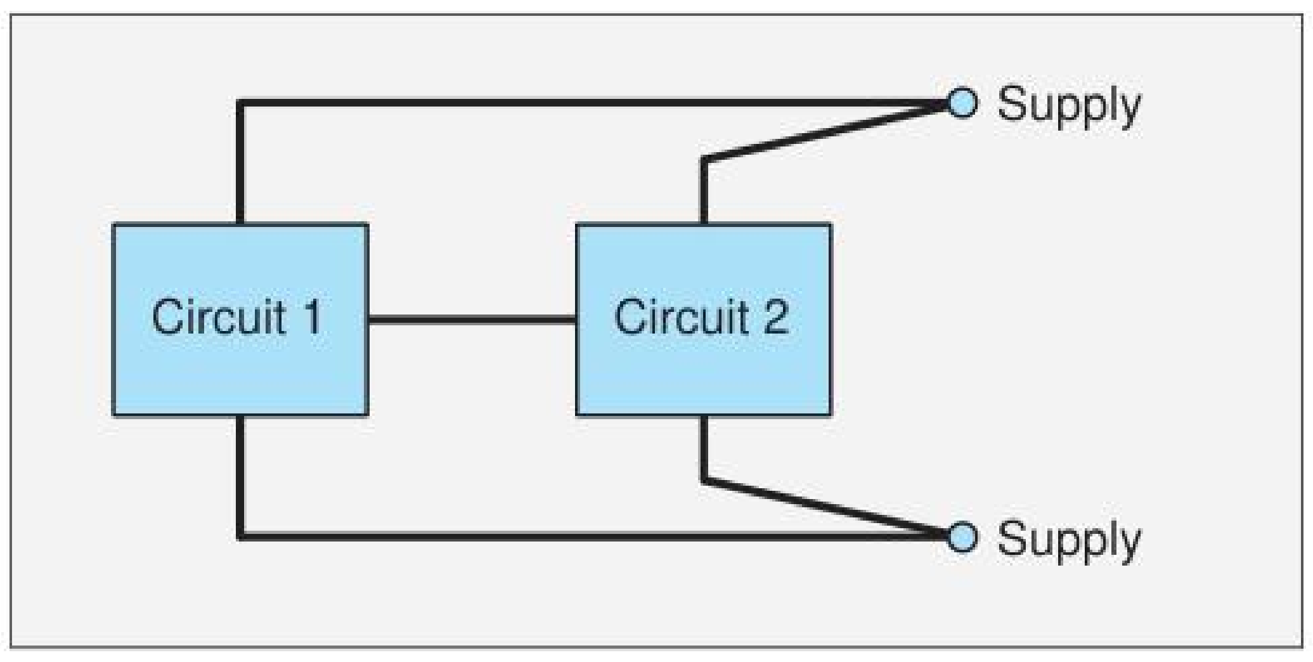


Fig.6. Supply wiring resistance causes voltage shifts and noise due to supply currents.

Fig.7. Supply wiring to reduce noise.



Ground loops

When two circuits, sub-circuits, instruments, or other equipment are grounded at two separate points on a 'ground bus' we have a situation know as a ground loop (or earth loop) (see Fig.8). The ground bus may be a circuit board track, the chassis of the equipment, point-to-point wiring, or the mains earth connected at different outlets – many people have suffered unnecessary levels of hum in their Hi-Fi systems due to earth loops!

The ground loop will pick up magnetic interference, probably mains hum and may also act like an antenna picking up radio frequency interference (RFI). Large loops will make the problem worse. Ground loops are a particular problem when two or more mains-powered systems (such as lab instruments and sensor circuits) are separately earthed and connected together. It is also possible for mains leakage currents to cause currents to flow in the earth (eg, shields) of connections linking equipment. Leakage currents can flow through parasitic capacitances and equipment ground, for example in transformers and EMI filters. The interference causes a current I_L to flow in the ground loop, which in turn causes an additional voltage drop (I_LR_G) across the resistance (R_G) of the ground connection between the equipment or subcircuits.

The solution to ground loops is to avoid them by using a single grounding point (Fig.8). Use of differential signals, only connecting screens at one end, use of very low resistance ground connections between circuits (reducing R_G), and signal isolation using transformers or opto-isolators also help minimise ground loop problems. Power isolation transformers may also help with mains wiring. Again, it is worth pointing out that some potential 'solutions' related to mains wiring, such as disconnecting earths, could be lethal.

Signal guarding

Signal guarding is concerned with getting the most out of screened cable connections, particularly when connecting very low-level signals from high-impedance sources to high-precision circuits. In such cases, effects such as leakage currents in the cables and cable capacitance can cause significant errors. Signal guarding uses triaxial cables and connectors (see Fig.9), which have an inner conductor, carrying the signal of interest and two layers of shielding. The inner shield is connected to a signal of equal voltage to the signal provided by a unity-gain amplifier (see Fig.10). This means that there is a zero-voltage difference between the signal and inner shield, so the leakage currents (and capacitance effects) are minimised. The outer shield is usually grounded and provides interference protection for the guard signal.

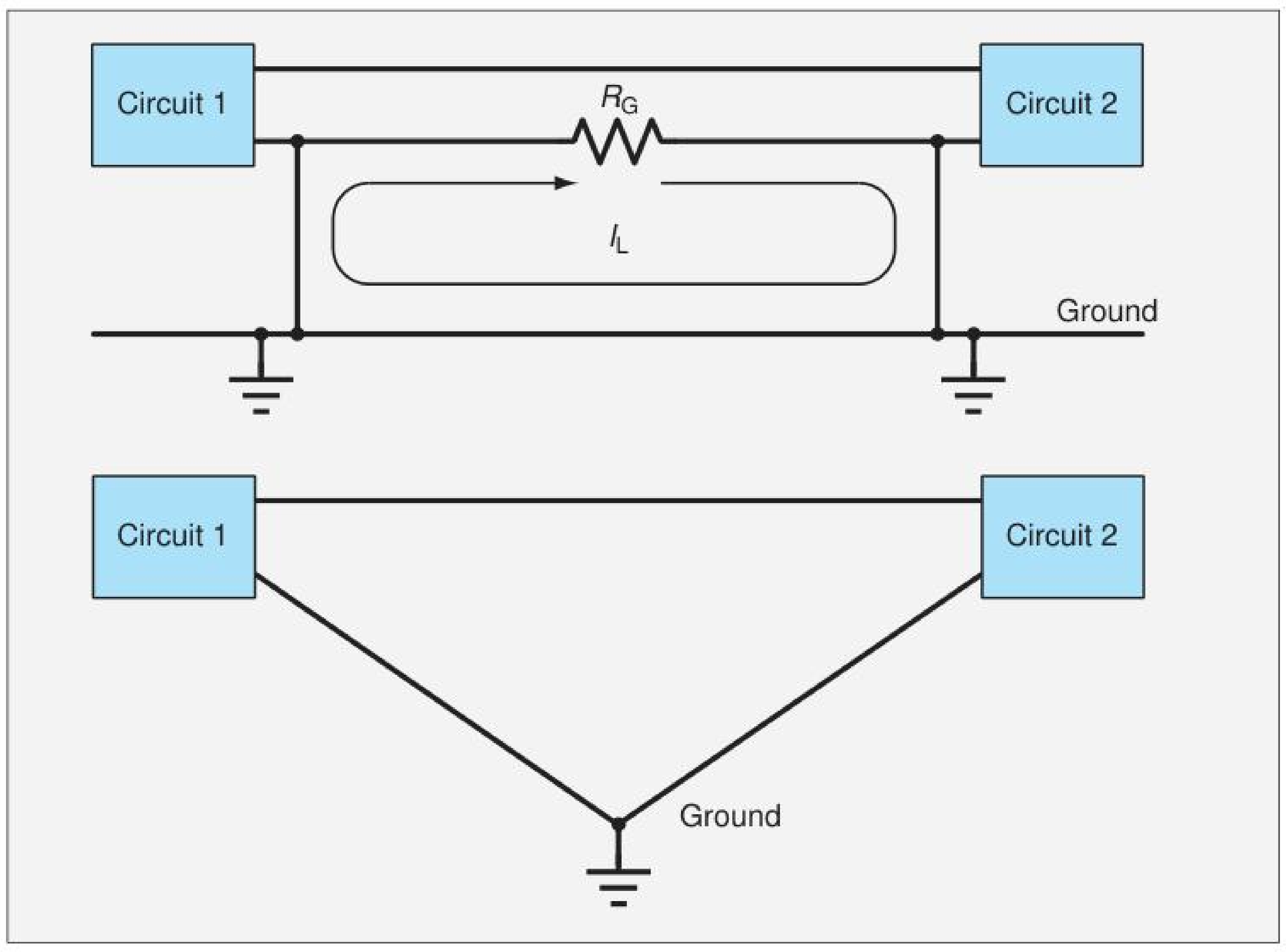


Fig.8. Ground loops: currents induced in ground loops cause voltage drops which introduce noise (upper schematic). Using a common ground point can eliminate the loop (lower schematic).

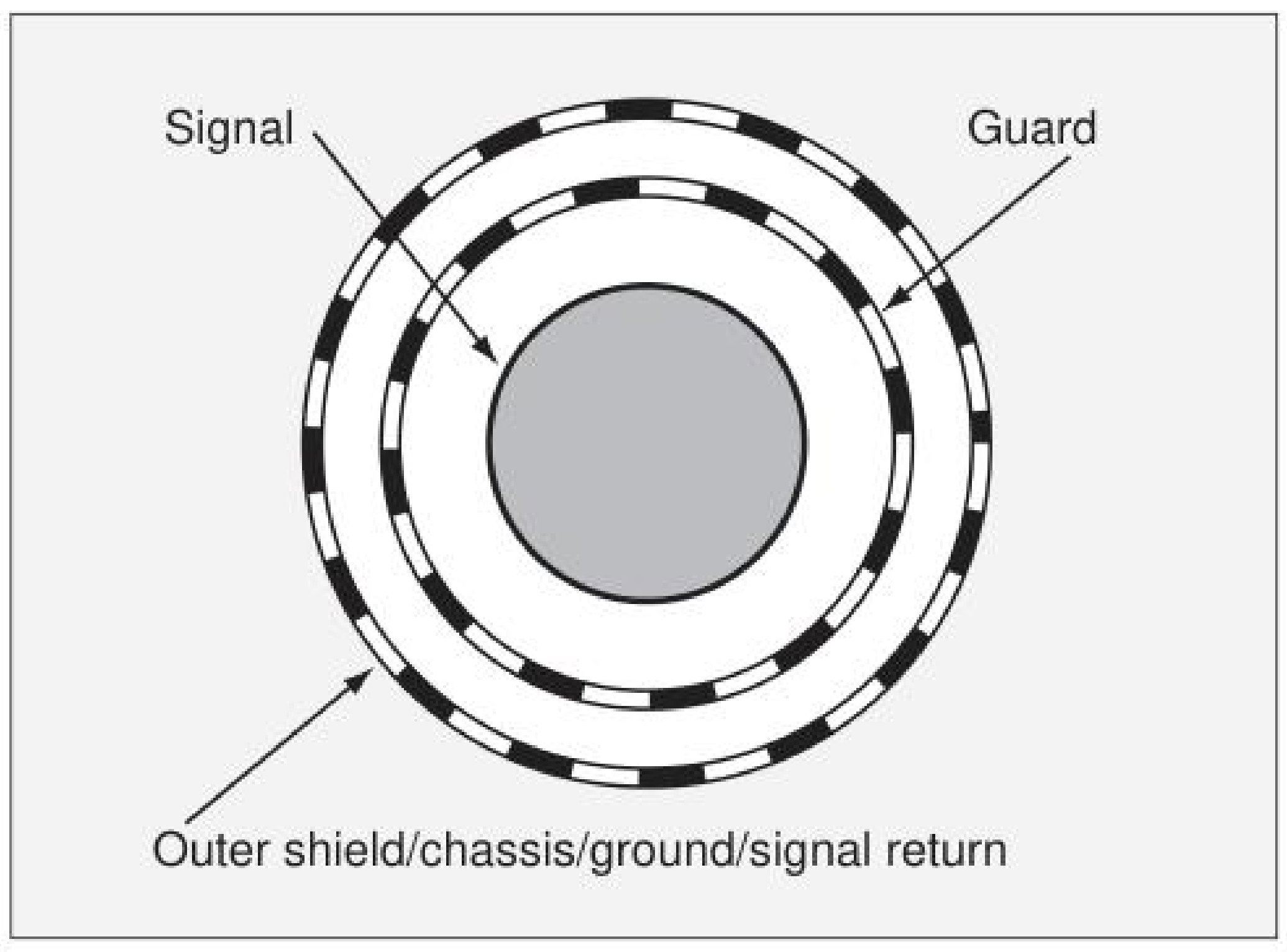


Fig.9. Triaxial connector used with triaxial cables for guarded connections.

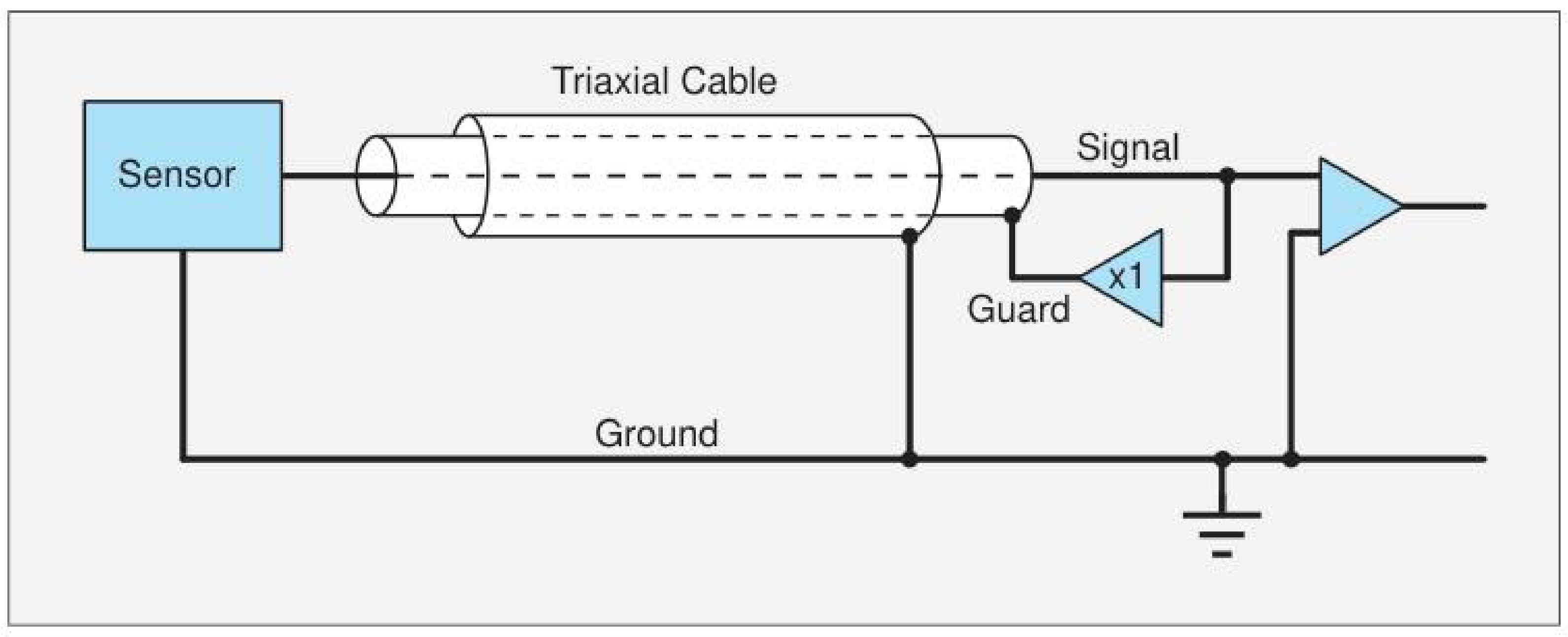


Fig.10. Guarded signal connection.

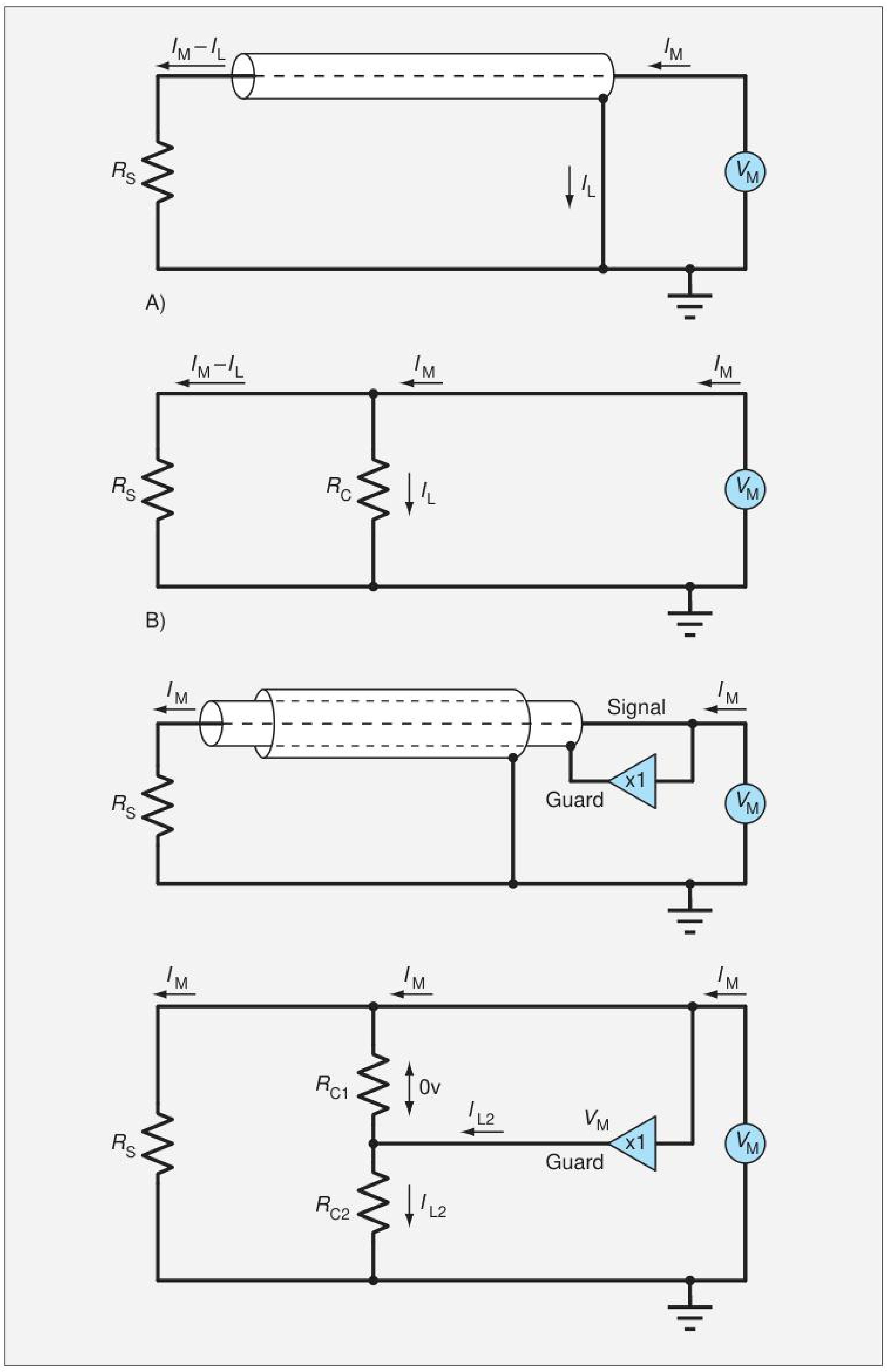


Fig.11. Guarded resistance measurement: a) non-guarded setup, b) non-guarded equivalent circuit, c) guarded setup, d) guarded equivalent circuit.

As an example of how guarding works, consider the schematic in Fig.11a, for which an equivalent circuit is shown in Fig.11b. Here we are trying to measure the resistance of a sensor (R_S) which has a very high resistance value and therefore leakage through the cable insulation resistance R_C is significant. We apply V_M and measure I_M – this should give the value of R_S as V_M/I_M, but if actually gives us this parallel combination of R_S and R_C due to the leakage current I_L. Using a guard (Fig.11c and Fig.11d) means that the voltage across R_{C1} between the inner conductor and guard is zero and hence no leakage current flows. The buffer amplifier has no difficulty in supplying the guard-to-ground leakage current I_{L2} and this does not disrupt the measurement.

Vibration and chemistry

Systems processing low-level signals are also prone to a variety of forms of interference-based noise and errors other than electrically/magnetically coupled signals, including mechanical and electrochemical effects. Movement and vibration of cables can create electric current through the triboelectric effect – charges created due to friction between a conductor and an insulator. Low-noise cables are available for situations where this may be a particular problem. Making sure that cables are well supported and not subject to vibration or large temperature fluctuations helps reduce this effect for any cable. Movement can also generate unwanted signals through the piezoelectric effect, which occurs when mechanical stress is applied to insulators. Unwanted signals due to movement and mechanical stress are sometimes called microphonic effects, because if the signal is listened to, the movement of (for example) a cable will be audible.

Batteries create electric current through electrochemical effects. Similar processes can occur if contaminants are present on PCBs and terminals. Variations in humidity can affect sensor systems with very high impedances. Contaminants (handling cables and connectors) can also reduce insulation resistance leading to errors.

ESR Electronic Components Ltd

All of our stock is RoHS compliant and CE approved. Visit our well stocked shop for all of your requirements or order on-line.

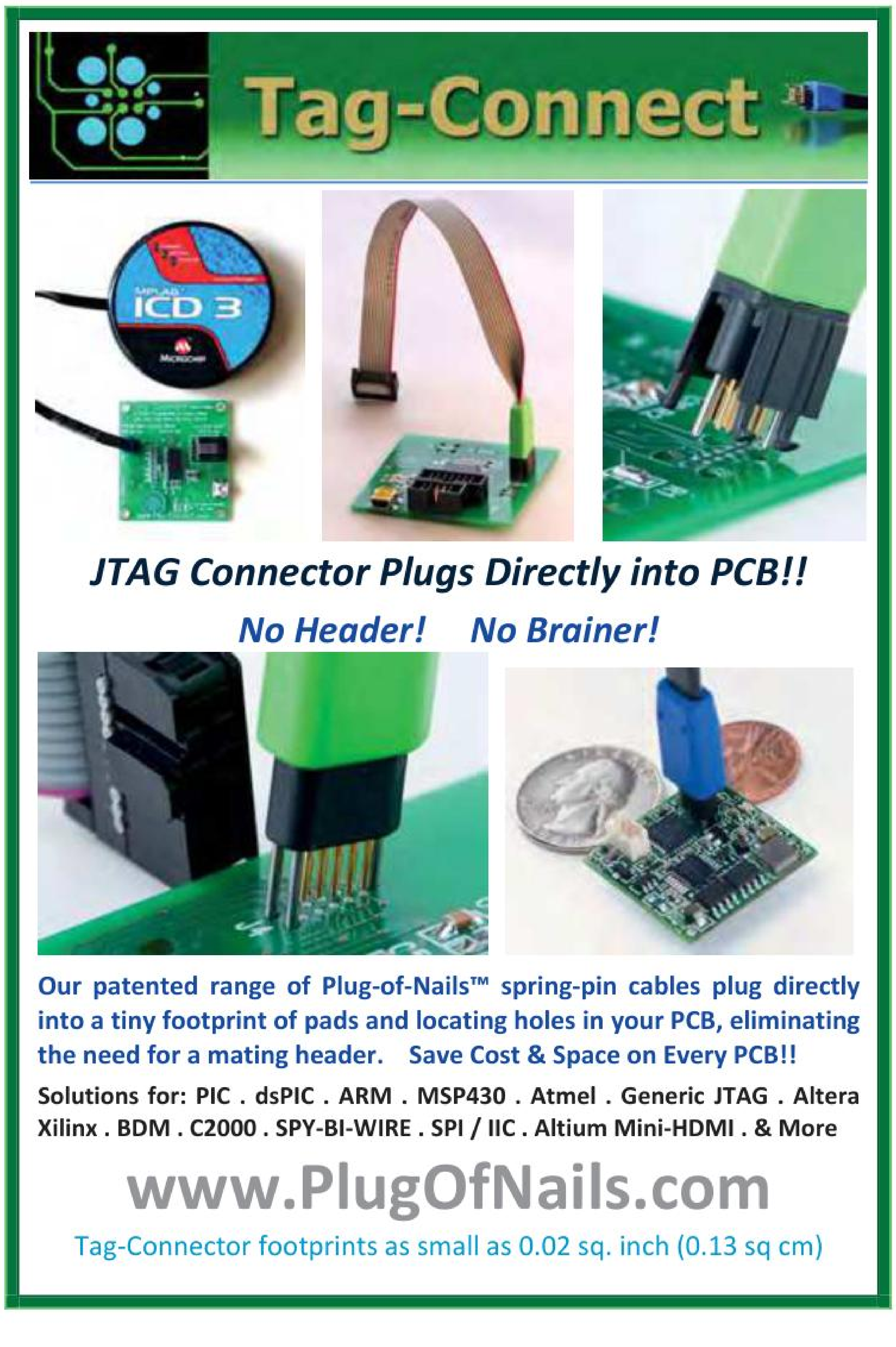
We can help and advise with your enquiry, from design to construction.

3D Printing • Cable • CCTV • Connectors • Components • Enclosures • Fans • Fuses • Hardware • Lamps • LED's • Leads • Loudspeakers • Panel Meters • PCB Production • Power Supplies • Relays • Resistors • Semiconductors • Soldering Irons • Switches • Test Equipment • Transformers and so much more…

Monday to Friday 08:30 - 17.00, Saturday 08:30 - 15:30

Station Road
Cullercoats
North Shields
Tyne & Wear
NE30 4PQ

Tel: 0191 2514363 sales@esr.co.uk www.esr.co.uk


Tag-Connect

JTAG Connector Plugs Directly into PCB!!

No Header! No Brainer!

Our patented range of Plug-of-Nails™ spring-pin cables plug directly into a tiny footprint of pads and locating holes in your PCB, eliminating the need for a mating header. Save Cost & Space on Every PCB!!

Solutions for: PIC . dsPIC . ARM . MSP430 . Atmel . Generic JTAG . Altera Xilinx . BDM . C2000 . SPY-BI-WIRE . SPI / IIC . Altium Mini-HDMI . & More

www.PlugOfNails.com

Tag-Connector footprints as small as 0.02 sq. inch (0.13 sq cm)

CRICKLEWOOD ELECTRONICS
Established 1981

Your best bet since MAPLIN
Chock-a-Block with Stock

Visit: www.cricklewoodelectronics.com

Or phone our friendly knowledgeable staff on **020 8452 0161**

Components • Audio • Video • Connectors • Cables Arduino • Test Equipment etc, etc

Visit our Shop, Call or Buy online at:
www.cricklewoodelectronics.com

020 8452 0161

Visit our shop at:
40-42 Cricklewood Broadway
London NW2 3ET

Max's Cool Beans

By Max the Magnificent

Flashing LEDs and drooling engineers – Part 11

Good grief! I am bubbling over with excitement. I have so many things I want to talk about that I don't have a clue where to begin. I know, I know... I need to sit down, take a deep breath, start at the beginning, work my way through the middle, and eventually stagger my way to the end. The problem is deciding what to talk about first. Ah, I can see in your eyes that you are desperate for me to commence with gamma correction. Well, if you insist...

Feeling off-colour

Way back in the mists of time when we started this mega-mini-series (*PE*, March 2020), we introduced the concept of pulse-width modulation (PWM). Since we can turn an LED on and off very quickly, the way we control its brightness is to vary the proportion of on time to its off time. We refer to this as the 'duty cycle.' A 20% duty cycle means the LED is on 20% of the time and off for the other 80%, while an 80% duty cycle means the LED is on 80% of the time and off the other 20%.

In the case of the microcontroller unit (MCU) we are working with – the Seeedunio XIAO – we can use values from 0 (0x00 in hexadecimal) being fully off (0% brightness) to 255 (0xFF in hexadecimal) being fully on (100% brightness). Using this PWM technique, a value of 128 (0x80 in hexadecimal) means the LED will be on half of the time and off the other half, resulting in 50% brightness.

In a later column (*PE*, September 2020), we introduced the colour wheel (Fig.1) that we decided to use for our 12×12 ping-pong array experiments (Fig.2). If you have built one of these arrays yourself, or if you are playing with tricolour LEDs in general, you may have noticed that some of the colours seem to be a tad 'off'. For example, the rose may appear very close to magenta, while the flush orange may appear more yellow that one might expect.

On the bright side (no pun intended), the red, green, and blue elements in our tricolour LEDs work as expected and provide a linear response such that a 50% duty cycle does indeed result in 50% brightness (Fig.2a). The problem is that our eyes have evolved to accommodate a huge dynamic range, from moonlight to sunlight and – as part of this – they have a sort of built-in non-linearity (Fig.2b).

Although not immediately obvious from my diagram, the curve of this non-linearity is defined by a somewhat tricky power-law function. In order to address this, we need to drive the red, green, and blue LEDs using the inverse of this function, which results in our eyes perceiving what we were hoping for in the first place (Fig.2c). We call the process of applying this inverse function, 'gamma correction.'

There's a great article on Adafruit's website covering all of this in depth (**https://bit.ly/31zaSLK**). As part of this, they provide what they call 'The Quick Fix' in the form of a cross-reference look-up table that we can use to remap the linear values we would like to use into their gamma-corrected counterparts that will provide us with the colours we want to see.

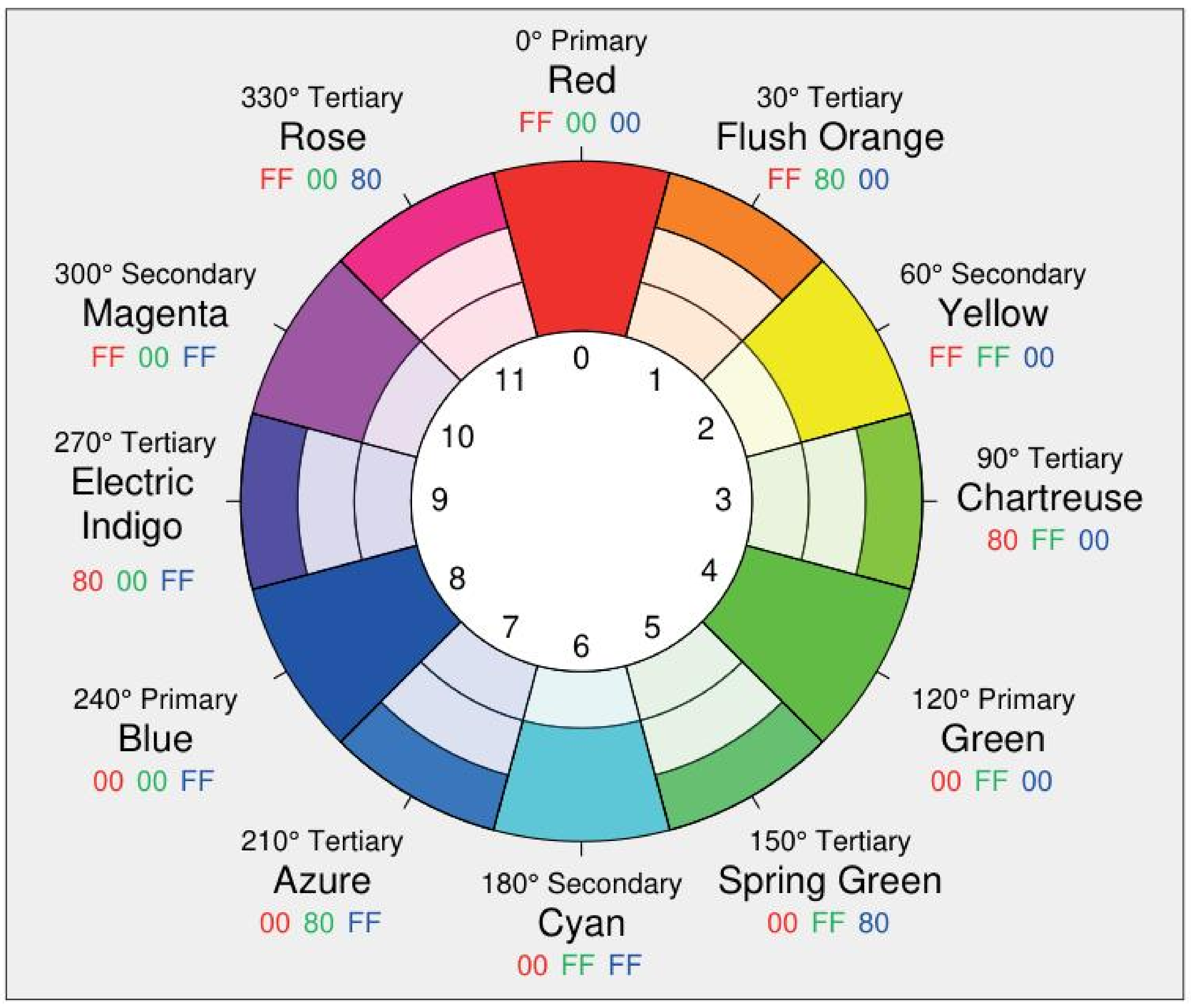
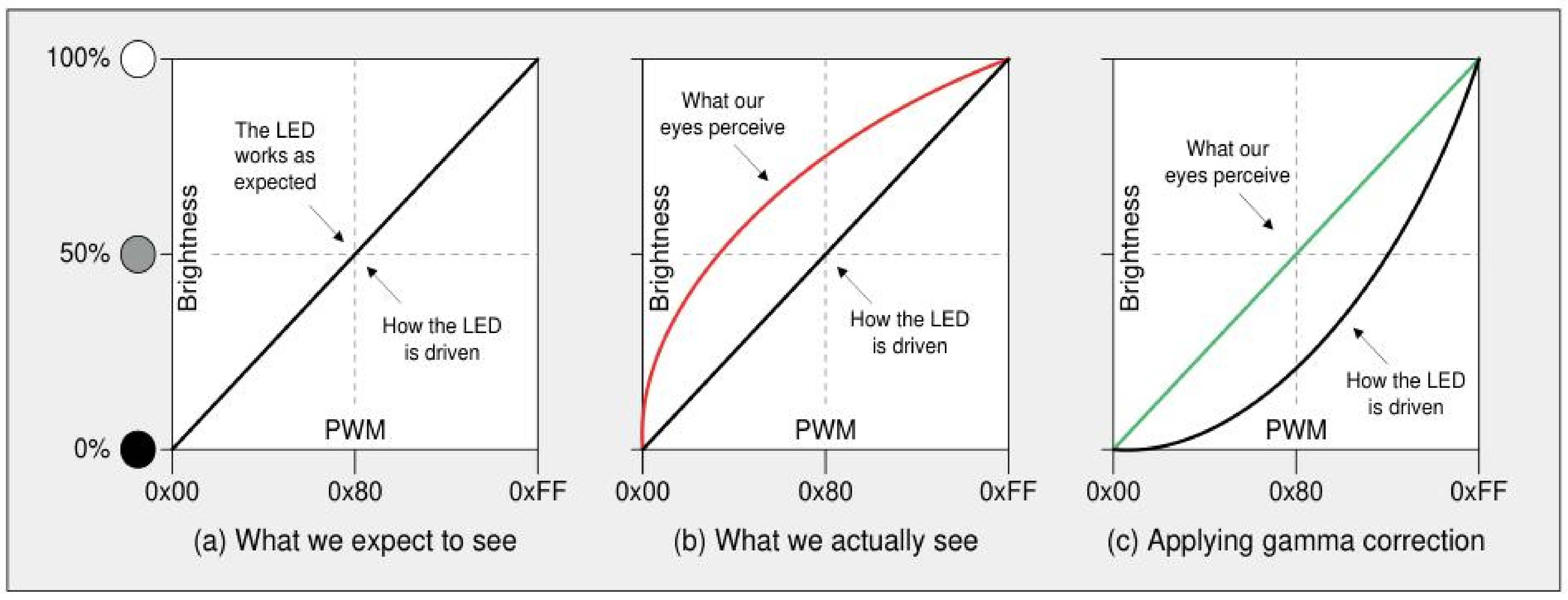


Fig.1. The colour wheel we've been using in our experiments.

I created a quick test sketch (program) to see how this would affect the colours on my 12×12 array (the full sketch is presented in file **CB-Jan21-01.txt** – it and the other files associated with this article, are available on the January 2021 page of the *PE* website). Since the folks at Adafruit were conscious that some of their users would be using low-end Arduinos with limited SRAM, they store their table in PROGMEM (the Flash program memory). By comparison, my Seeedunio XIAO has so much memory that I can afford to flaunt it, so I dispatched the butler to fetch my flaunting trousers and stored my table in SRAM (Fig.3.).

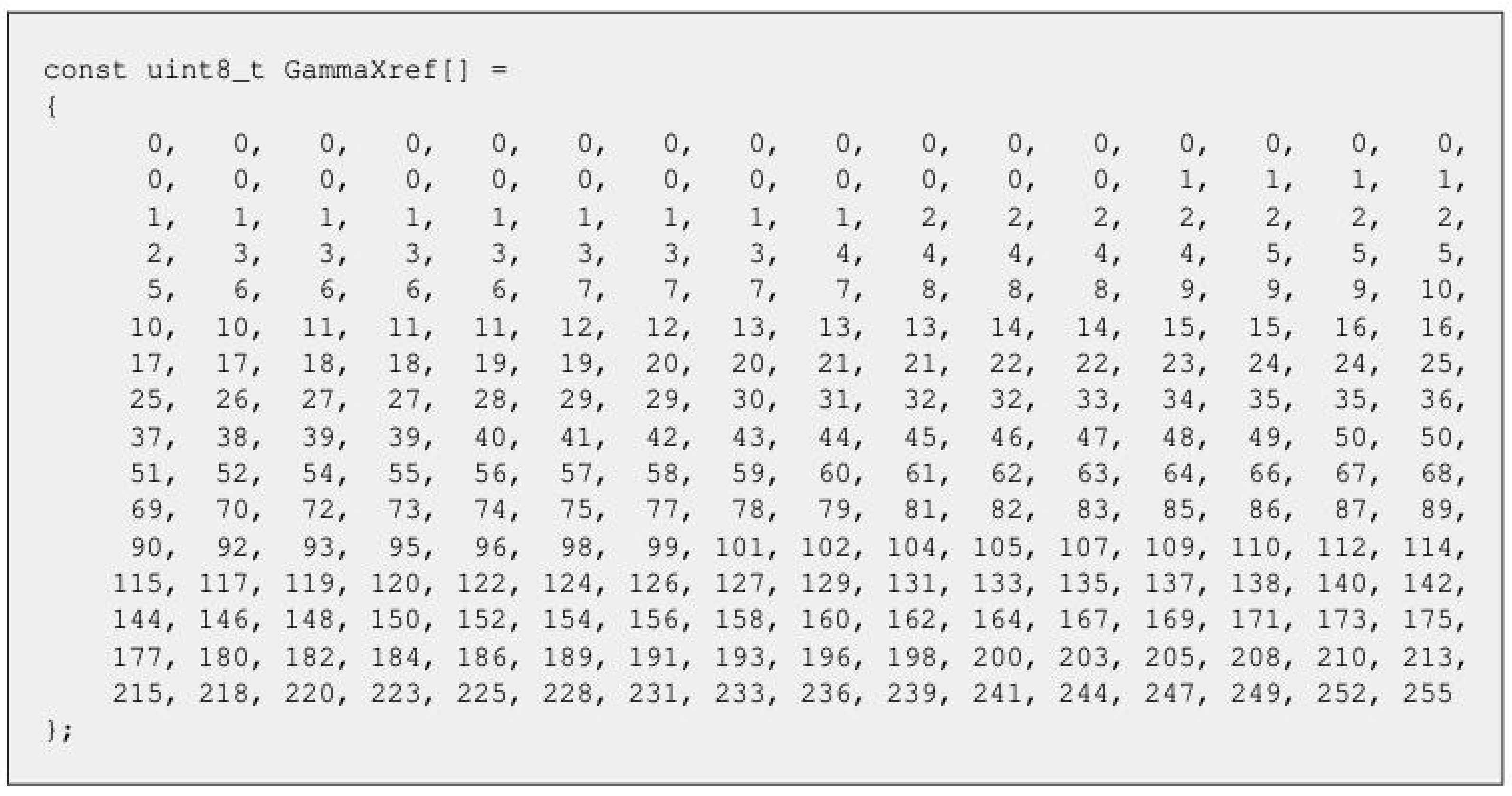
Previously (*PE*, October 2020), we met the `GetRed()`, `GetGreen()`, and `GetBlue()` functions that extract and return the 8-bit red, green, and blue components from a 32-bit colour value. Also, we introduced the `BuildColor()` function that accepts 8-bit red, green, and blue components and returns a 32-bit colour value.

The thing is, we have to apply our gamma correction to each of the colour channels individually. Thus, in our new sketch, we've added a `GetGammaCorrectedColor()` function that accepts a 32-bit colour value, splits it into its red, green and blue components, uses our `GammaXref[]` look-up-table to apply gamma correction to each component, and then returns the gamma-corrected 32-bit result.



(a) What we expect to see (b) What we actually see (c) Applying gamma correction

Fig.2. Gamma correction.



```
const uint8_t GammaXref[] =
{
      0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
      0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   1,   1,   1,   1,
      1,   1,   1,   1,   1,   1,   1,   1,   1,   2,   2,   2,   2,   2,   2,   2,
      2,   3,   3,   3,   3,   3,   3,   3,   4,   4,   4,   4,   4,   5,   5,   5,
      5,   6,   6,   6,   6,   7,   7,   7,   7,   8,   8,   8,   9,   9,   9,  10,
     10,  10,  11,  11,  11,  12,  12,  13,  13,  13,  14,  14,  15,  15,  16,  16,
     17,  17,  18,  18,  19,  19,  20,  20,  21,  21,  22,  22,  23,  24,  24,  25,
     25,  26,  27,  27,  28,  29,  29,  30,  31,  32,  32,  33,  34,  35,  35,  36,
     37,  38,  39,  39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  50,
     51,  52,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,  66,  67,  68,
     69,  70,  72,  73,  74,  75,  77,  78,  79,  81,  82,  83,  85,  86,  87,  89,
     90,  92,  93,  95,  96,  98,  99, 101, 102, 104, 105, 107, 109, 110, 112, 114,
    115, 117, 119, 120, 122, 124, 126, 127, 129, 131, 133, 135, 137, 138, 140, 142,
    144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 167, 169, 171, 173, 175,
    177, 180, 182, 184, 186, 189, 191, 193, 196, 198, 200, 203, 205, 208, 210, 213,
    215, 218, 220, 223, 225, 228, 231, 233, 236, 239, 241, 244, 247, 249, 252, 255
};
```

Fig.3. Gamma-correction cross-reference look-up table.

```
uint32_t GetGammaCorrectedColor
              (uint32_t uncorrectedColor)
{
  uint8_t  tmpRed;
  uint8_t  tmpGreen;
  uint8_t  tmpBlue;

  uint32_t gammaCorrectedColor;

  tmpRed = GammaXref[ GetRed(uncorrectedColor) ];
  tmpGreen = GammaXref[ GetGreen(uncorrectedColor) ];
  tmpBlue = GammaXref[ GetBlue(uncorrectedColor) ];

  gammaCorrectedColor = BuildColor(tmpRed,
              tmpGreen, tmpBlue);

  return gammaCorrectedColor;
}
```

The rest of the sketch is used to load the left-hand side of the array with uncorrected colours directly from our colour wheel and the right-hand side with their gamma-corrected counterparts, starting with red at the bottom and ending with rose at the top (Fig.4).

Although it's not easy to see from this image, in the real world the gamma-corrected values do present what appears to be a richer colour palette. For example, the gamma-corrected orange (second row from the bottom) looks more orange and the gamma-corrected rose (top row) appears more vibrant.

War, what is it good for?

According to Edwin Starr in his 1970 hit single 'War' – and Frankie Goes to Hollywood more than a decade later – the answer is 'Absolutely nothing.' Of course, it may be that neither of these luminaries were familiar with the concept of 'Code War'.

Actually, if the truth be told, neither was I until my chum, Ken Wood, who has been following these columns, sent me an email telling me all about this idea. In 1984, Alexander Dewdney wrote a column in *Scientific American* magazine about a programming game called *Core War* that he had created with DG Jones. A scan of this original article, along with a lot of supporting material, can be found on the **CoreWars.org** website and Wikipedia.

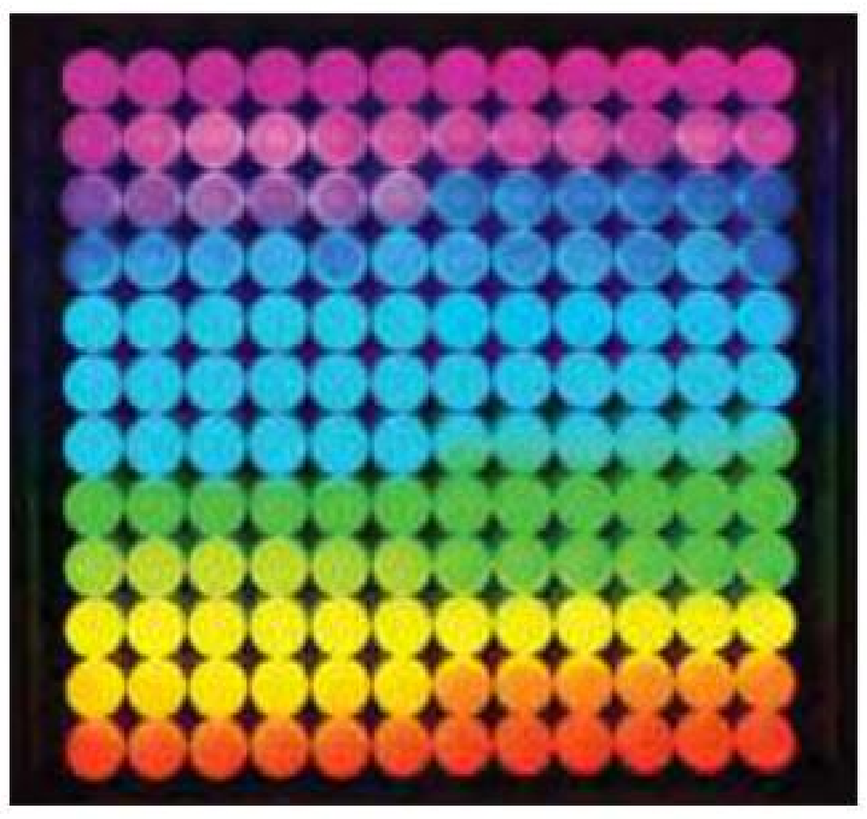


Fig.4. Uncorrected colours (left) vs. gamma-corrected colours (right).

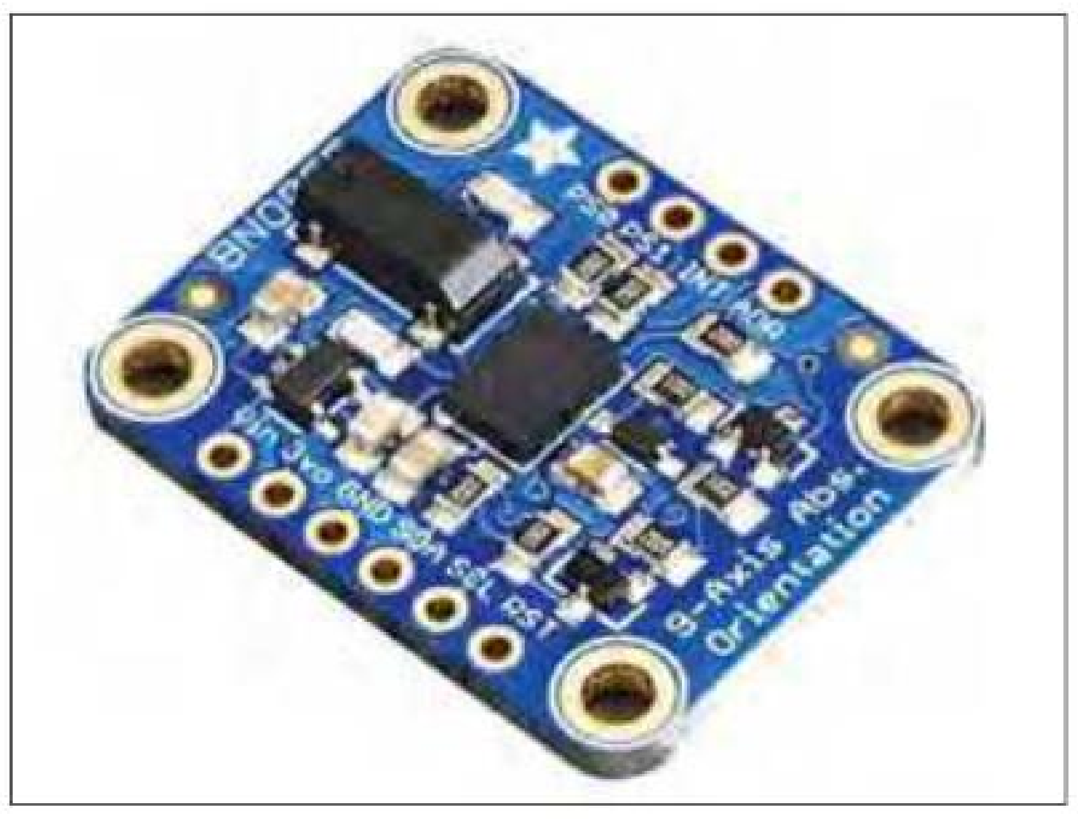


Fig.5. 9DOF BoB (Image source: Adafruit.com)

The idea is that you have two programs fighting each other in a virtual machine known as the Memory Array Redcode Simulator (MARS). The objective is to be the last program standing. To that end, each program can try to sabotage the other one and/or try to defend itself by self-repairing. You can get a really good feel as to what this is all about by reading the *Beginner's Guide* to the Redcode pseudo assembly language that is used to create the warrior programs (**https://bit.ly/34m5YUo**).

As Ken said in his email, 'I figure this can be made visually appealing by presenting the memory array on a screen (or ping-pong ball array) and colour-coding each cell either 'Neutral,' 'Last written for or by Program A,' or 'Last written for or by Program B' — using green, blue, and red respectively, for example — and running the programs at only a few steps per second so progress can be followed.'

Initially, I was a tad skeptical that a 144-element MARS would suffice but – having looked at the Redcode *Beginner's Guide* – I've changed my mind. Now I'm thinking about creating a MARS simulator to run on the Seeeduino XIAO that I'm using to power my 12×12 array. I'm also thinking about creating a Redcode assembler utility that can generate the warrior programs to run on the simulator.

But wait, there's more! Do you remember me talking about the *NeoPixel Simulator* that you can use to test your own programs to run on my 12×12 array (*PE*, November 2020)? Well, if I manage to find the time to get a MARS simulator up and running, we could combine it with our *NeoPixel Simulator*, thereby allowing you to create and test your own *Code War* warrior programs and then send them to me to be run on the real array.

Keep your balance

When I was a kid, my parents bought me a wooden marble maze toy. I just found something quite similar on Amazon (**https://amzn.to/2HwZg4M**), although the one I owned had larger mazes and used smaller ball bearings.

The reason I mention this here is that a reader emailed me to suggest I attach a sensor to my 12×12 array such that, if the array is held in a horizontal plane, I could control the 'rolling' of a lit pixel by detecting the tilt of the array.

By some strange quirk of fate, I just happened to have one of Adafruit's BNO055-based 9DOF (nine degrees of freedom) Fusion breakout boards (BOBs) in my treasure chest (junk box) of spare parts (Fig.5) (**https://bit.ly/3dP8EwU**).

This little beauty is based on a BNO055 microelectromechanical system (MEMS) sensor from Bosch. In turn, the BNO055 contains a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer (they also throw in a temperature sensor for good measure). The really cool thing about this device is that it also contains a 32-bit Arm Cortex M0+ processor, which performs all sorts of mindbogglingly complicated sensor algorithms for you and provides you with data in a form you can use without your brains leaking out of your ears.

As usual, the folks at Adafruit provide a wealth of information on this sensor, including pinouts, wiring, and how to download the required libraries (**https://bit.ly/35sVvpz**). Also included is some sample Arduino Code, which I used to create my first test program. The purpose of this initial sketch was to make sure I could get my XAIO microcontroller to talk to the BNO055. All we do is loop around reading the *x*, *y*, and *z* orientation values from the BNO055 and display them as floating-point values on the Arduino's Serial Monitor. Note that the XIAO communicates with the BNO055 via an I^{2}C bus, which uses pins 5 and 6 on the XIAO, but we don't declare these pins in our sketch because Adafruit's libraries handle all of this for us (file **CB-Jan21-02.txt**).

The next step involved some mental gymnastics to visualize how I was going to mount my breadboard in the 12×12 array case, and which (sensor) values corresponded to what (left-right and forward-backward) tilts. Eventually, I determined that the *y* values from the sensor would reflect tilting the array to the left or right, while the *z* values from the sensor would reflect tilting the array to the front or back. Thus, my second test program involved my taking the floating-point *y* and *z* values from the sensor and converting them into integer tilt values for use in my sketch.

With a little manipulation, I ended up with an integer `tiltLeftRight` value that ranges from –90 (tilted left so much as to be vertical), through 0 (horizontal), to +90 (tilted right so much as to be vertical). Similarly, I ended up with an integer `tiltFrontBack` value that ranges from –90 (tilted down to the front – that is, to the user – so much as to be vertical), through 0 (horizontal), to +90 (tilted to the back so much as to be vertical). In this case, all we do is loop around reading the *y* and *z* orientation values from the BNO055 and display them as my manipulated integer values on the Arduino's Serial Monitor (file **CB-Jan21-03.txt**).

Now things start to get more interesting. In our next sketch, we commence by setting one of the pixels – our 'ball' – in the center of the array to white. We then loop around using the readings from the sensor to make this pixel 'roll around' in response to tilting the array. The 'ball' stops 'rolling' when it hits one of the sides or gets trapped in one of the corners until we tilt the array in the opposite direction (file **CB-Jan21-04.txt**). For your delectation and delight, I just captured a short video showing this program in action (**https://bit.ly/3mXHZS3**).

Of course, this is only the beginning. Currently, the 'ball' simply 'rolls' at a constant rate once the tilt has passed a certain threshold. The next step will be to add some physics into the mix such that the speed of the 'roll' varies as a function of the angle of the tilt.

After that, we are limited, as usual, only by our own imaginations. For example, we could make green 'food' pixels randomly appear on the array and then try to guide our rolling pixel to hit them and 'eat' them before they randomly disappear again. The more you 'eat,' the more points you get. Also, we could add red 'hole' pixels into which our rolling pixel would 'fall' if they were to come into contact (these 'hole' pixels could be stationary, or we could make them randomly appear and disappear like the 'food' pixels).

We could come up with all sorts of cool games based on this technology if we put out heads together. Speaking of which, if you think of anything, please drop me an email and share the good word.

Size does matter!

To paraphrase an old saying, 'It's not the size of your array, it's what you do with it that counts.' In reality, of course, we all know in our heart of hearts that size does indeed matter. If you have two people standing next to each other, one clutching a paltry array and the other staggering under the weight of a gargantuan offering, which one do you think the punters are going to be looking at?

'Great minds think alike,' as they say. Of course, they also say that 'Fools seldom differ,' but I'm sure that doesn't apply to us. For example, a guy who goes under the moniker 'Bitluni' created a 20×15 = 300 ping-pong ball display using tricolour LEDs (**https://bit.ly/3jmltjj**). Sometime later, the attention-grabbing Bitluni followed up with a 40×30 = 1200 ball array (**https://bit.ly/31Bv8g0**).

In an earlier column, when I was building my 12×12 array, I noted that if I ever did this again, I would build 8×8 sub-arrays and then link them together to form a larger array. Well, the over-achieving Bitluni is of like mind, because he recently created a 48×40 = 1920 ball array using this very technique (**https://bit.ly/3jrwKyV**).

I can think of so many things we could do with such an array. For example, can you imagine using a machine vision system based on artificial intelligence (AI) to detect and recognize gestures as we wave our arms around controlling a wall-size version of the game Tetris? All I can say is that I hope to meet up with the nefarious Bitluni one day (and appropriate his wall-sized array while he's not looking).

Prognostication revisited

I don't know if you recall (I can barely remember myself), but this entire saga commenced with my wishing to illuminate the controls on my *Pedagogical and Phantasmagorical Inamorata Prognostication Engine*. In addition to lighting the furnace in the upper control panel and the ginormous vacuum tubes sitting on top of the engine, the main control panel features eight toggle switches and six pushbuttons, each of which is accompanied by two tricolour LEDs. Also, there are five potentiometers (pots), each surrounded by a ring of 16 tricolour LEDs (Fig.6).

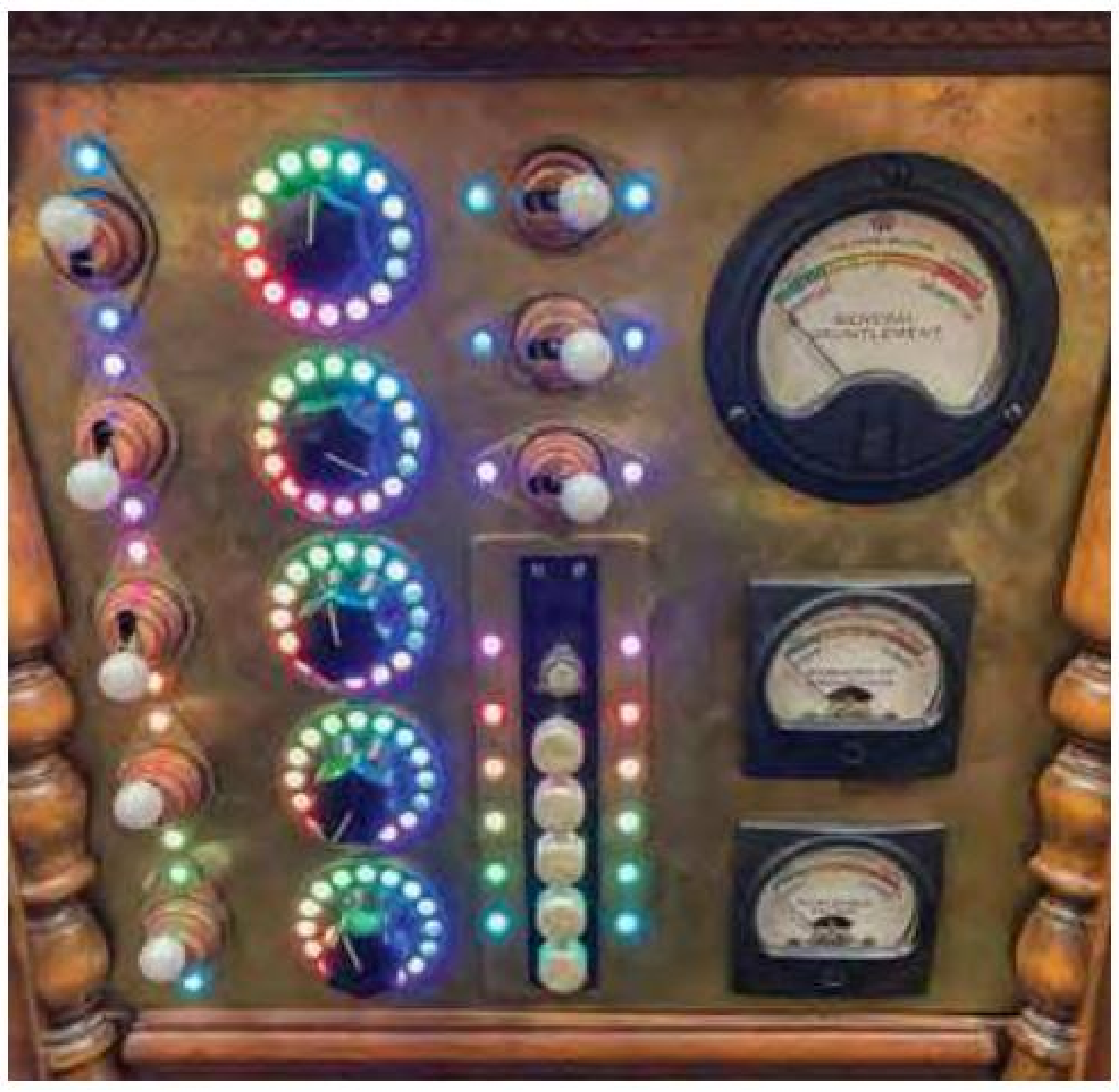


Fig.6. The *Prognostication Engine's* main control panel.

We've discussed some of the effects we could employ with the switches and pushbuttons in previous columns, so let's now turn our attention to the LEDs associated with the pots.

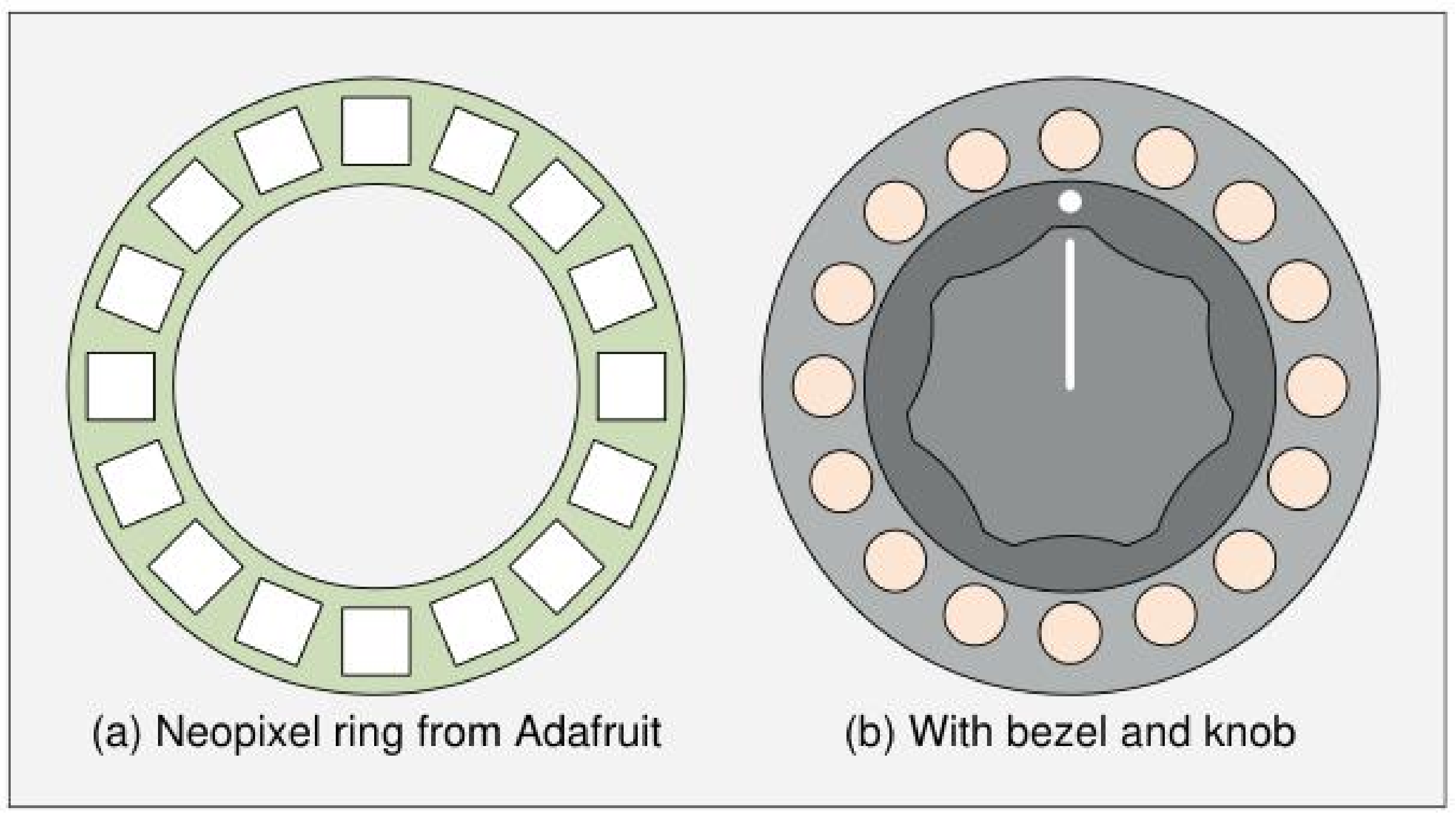
As an aside, the pots are motorised, so if some unauthorised person were to attempt to change the engine's settings, as soon as they go 'hands-off,' the pots will automatically return to their officially designated positions under program control.

Remember that we are using 16-element NeoPixel rings from Adafruit (**https://bit.ly/37RtFpQ**). In front of each ring (Fig.7a) we have a brass bezel and an antique Bakelite knob (Fig.7b). Mounted in the bezel, in front of each pixel, we have a small pseudo-mother-of-pearl 'dot' which adds to the steampunk look-and-feel.

Since we have 16 pixels, we can consider each as spanning (or representing) an arc of 360°/16 = 22.5°. Now, different pots have different physical ranges for how far they can rotate. The ones I'm using support rotations of 290°, which means there are three pixels to which the pointer on the knob cannot, in fact, point (Fig.8).

Of course, although these pixels are shown as being dark gray in the figure, the fact that we can't point to them doesn't mean we can't light them up. One option would be to simply paint the white line and dot on the knob black, and then map (translate) the 290° rotation of the knob into a 360° range on the ring.

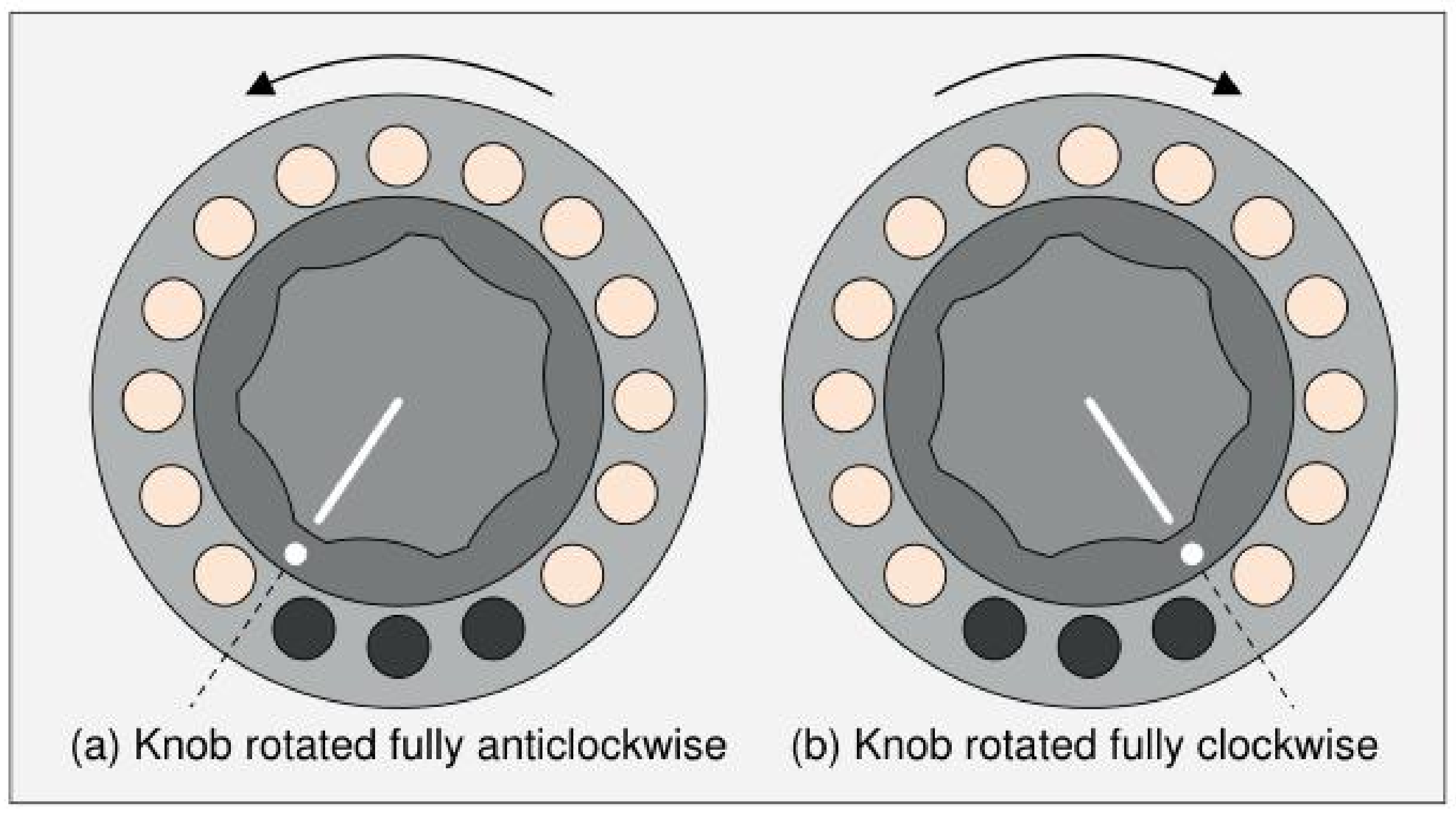
However, I like the white line and dot on the knob. We could just turn the three



(a) Neopixel ring from Adafruit

(b) With bezel and knob

Fig.7. 16-pixel rings.



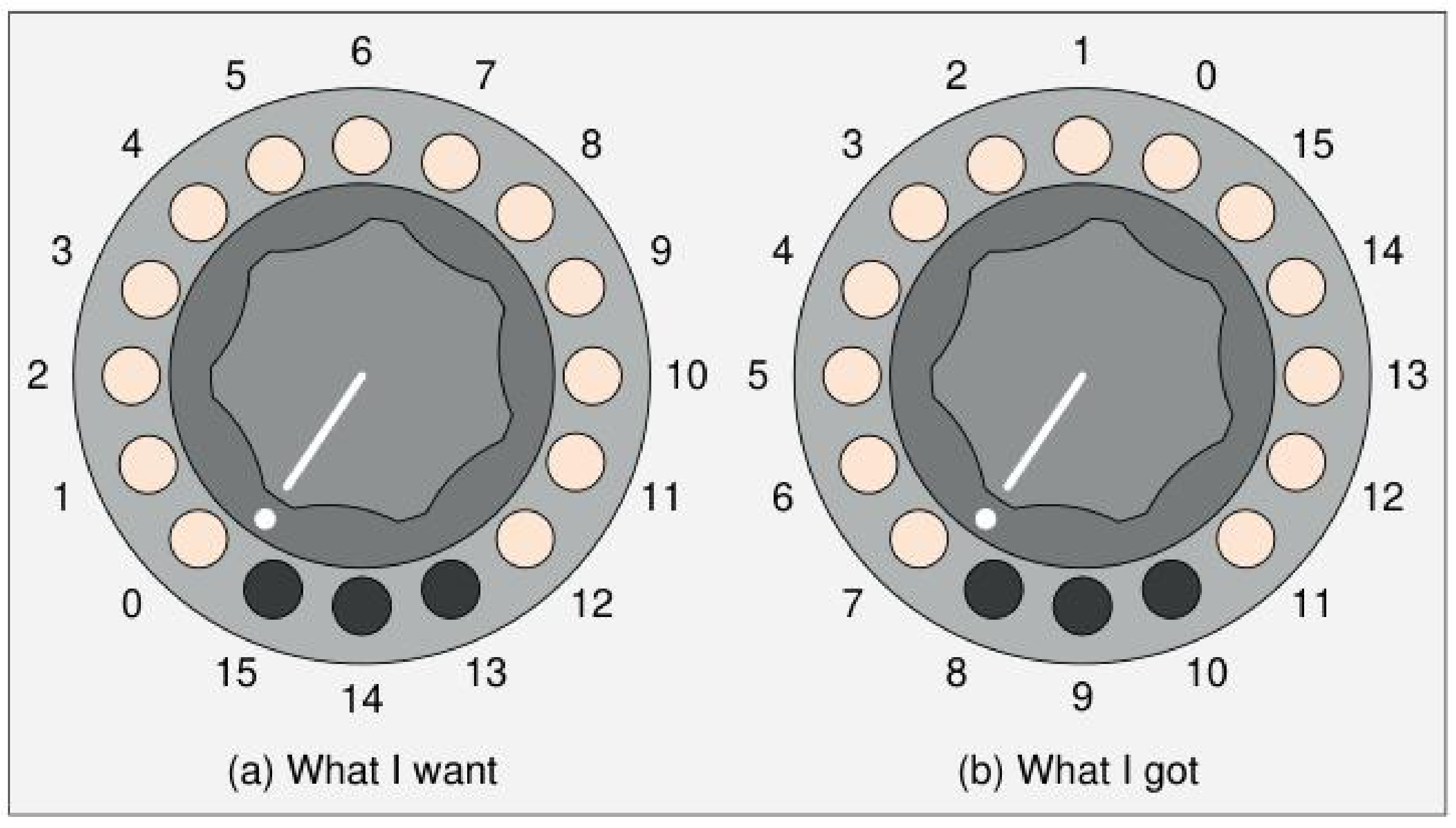
(a) Knob rotated fully anticlockwise

(b) Knob rotated fully clockwise

Fig.8. The rotation of the knob.

pixels at the bottom off, but where would be the fun in that? My solution will be to use a separate colour for these three pixels. Initially this will just be a steady colour, but it may be that later we use different colours, fades, and flashes on these three pixels to provide us with additional information about the state of the machine.

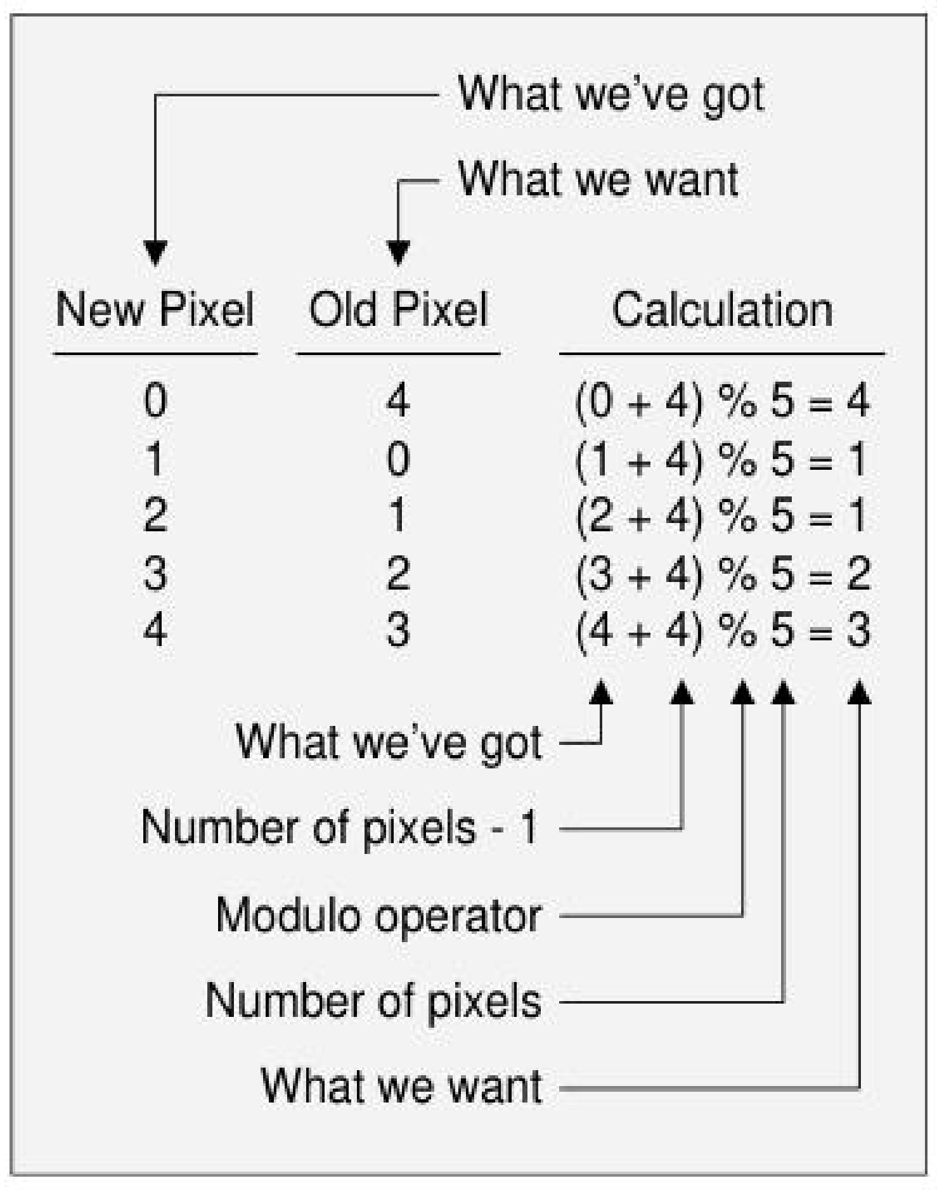
The way I want to visualise things is with pixels 0 and 12 corresponding to the maximum anticlockwise and clockwise rotations of the knob, respectively (Fig.9a). However, the way in which I created my prototype resulted in the pixels being presented in a different manner (Fig.9b).



(a) What I want

(b) What I got

Fig.9. You can't always get what you want.



New Pixel	Old Pixel	Calculation
0	4	(0 + 4) % 5 = 4
1	0	(1 + 4) % 5 = 1
2	1	(2 + 4) % 5 = 1
3	2	(3 + 4) % 5 = 2
4	3	(4 + 4) % 5 = 3

Fig.10. Thought experiment with 5-pixel ring.

To be honest, I don't recall how the rings are oriented on the main *Prognostication Engine*, but it really doesn't matter because we are going to perform a simple cross-reference operation, and we can easily modify the cross-reference values later. In the case of our prototype, we will declare our cross-reference values as follows:

```
int RingXref[NUM_NEOS_RING] =
      {7,6,5,4,3,2,1,0,15,14,13,12,11,10,9,8};
```

The way this works is really simple, if we want to light what we like to think of as pixel `i` in our imaginary world, we will access `RingXref[i]`, which will return the number of the corresponding pixel in the real world; for example, `RingXref[0]` will return 7 (tra-la!).

So, just to illustrate where we're at thus far, what we are going to do is create a simple sketch that lights our pixels from 0 to 15, first red, then blue, then green, then start all over again. Since these LEDs are so bright, we're going to add a `ModifyBrightness()` function that will dim them down to a specified percentage of their full value, which will allow us to keep our original colour definitions as-is. The heart of this program is a function called `LightMultiple()`, which we will call from our `loop()` function (file **CB-Jan21-05.txt**).

```
void LightMultiple (uint32_t thisColor)
{
   int      tmpNeo;
   uint32_t tmpColor;

   tmpColor = ModifyBrightness(thisColor, BRIGHTNESS);

   for (int iNeos = 0; iNeos < NUM_NEOS_RING; iNeos++)
   {
      tmpNeo = RingXref[iNeos];
      NeosRing.setPixelColor(tmpNeo, tmpColor);
      NeosRing.show();
      delay(InterPixelPadDelay);
   }
}
```

As you can see, we first modify the brightness of the colour (I've cut things down to only 20% for this experiment, and they are still bright). Next, we cycle round lighting each of our pixels, using our `RingXref[]` to cross-reference the pixel numbers, upload the new values, and pause for a short delay that we've called `InterPixelPadDelay`. If you look at the code, you'll see that I've defined a `CYCLE_TIME` of one second. Also, I'm assuming that it takes only 1ms (one millisecond, represented by `CALC_UPLOAD_DELAY`) to perform all my calculations and upload the new values. If you look at the `setup()` function, you'll see how we used these values, along with the number of pixels in the ring, to calculate the `InterPixelPadDelay`.

Still prognosticating furiously

Although lighting the pixels the way we've just done won't form part of the *Prognostication Engine's* primary function, it helps us to wrap our brains around some of the nitty-gritty details, and we will be able to use all of this stuff as part of a flamboyant power-up display.

In this vein, suppose we want to modify our previous program such that we have only a single pixel lit at any one time. In this case, when we turn our new pixel on, we also want to turn the previous pixel off. Remembering that we are cycling the pointer to our pixels, `iNeo`, from 0 to 15, we would ideally like to modify our core `for()` loop do something like the following (the new code is shown in bold):

```
for (int iNeos = 0; iNeos < NUM_NEOS_RING; iNeos++)
{
   // Turn the new pixel on
   tmpNeo = RingXref[iNeos];
   NeosRing.setPixelColor(tmpNeo, tmpColor);

   // Turn the old pixel off
   tmpNeo = iNeos - 1;
   tmpNeo = RingXref[tmpNeo];
   NeosRing.setPixelColor(tmpNeo, COLOR_BLACK);

   NeosRing.show();
   delay(InterPixelPadDelay);
}
```

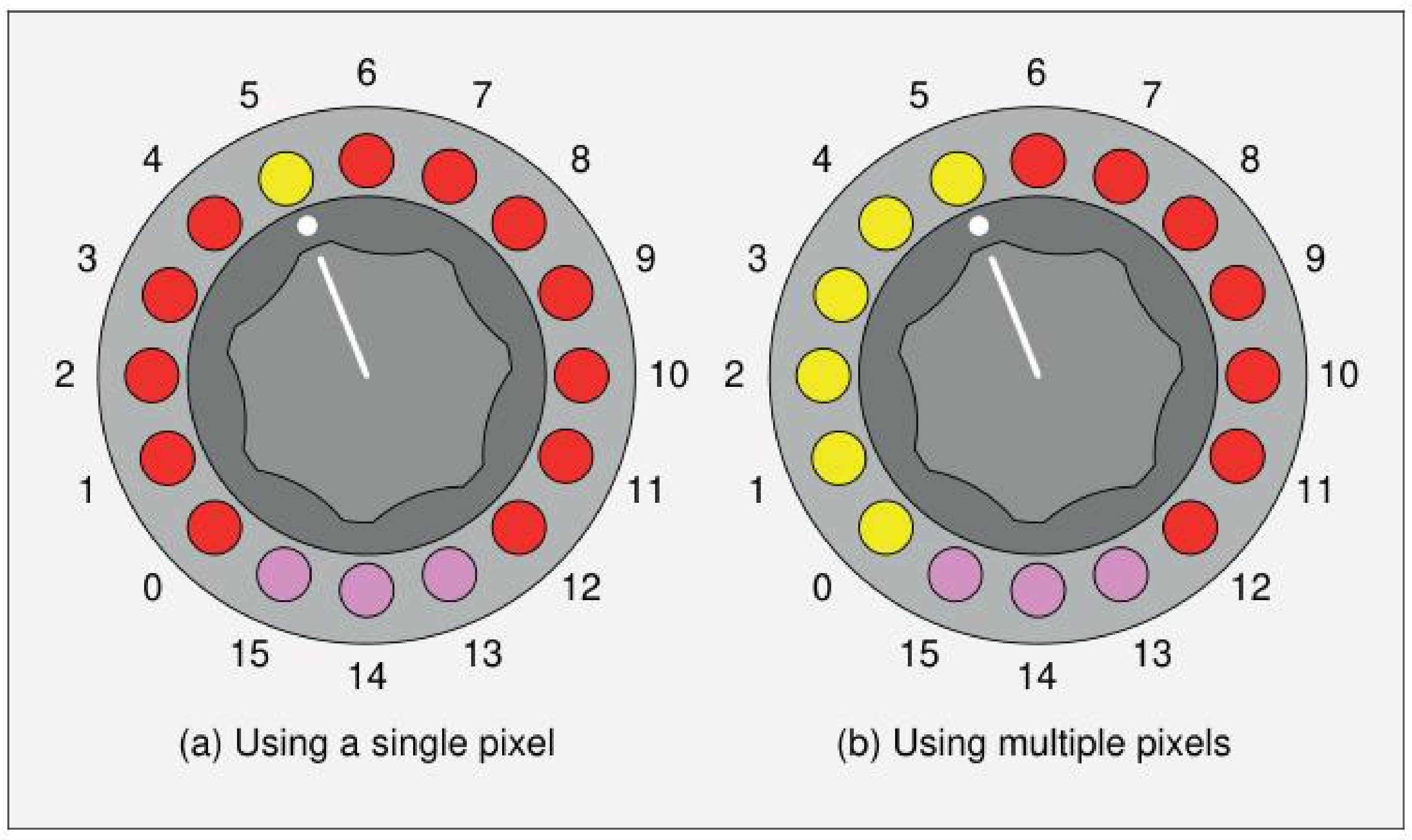
This would work great for every value of `iNeo` except 0 because 0 – 1 = –1, which will cause our program to do something excruciatingly painful when we attempt to use this value as an index into `RingXref[]`. It's always the 'end conditions' that bite you when you are least expecting it. One alternative would be to add a test for this end condition as shown below:

```
for (int iNeos = 0; iNeos < NUM_NEOS_RING; iNeos++)
{
   // Turn the new pixel on
   tmpNeo = RingXref[iNeos];
   NeosRing.setPixelColor(tmpNeo, tmpColor);

   // Turn the old pixel off
   if (iNeos == 0) tmpNeo = 15; else tmpNeo = iNeos - 1;
   tmpNeo = RingXref[tmpNeo];
   NeosRing.setPixelColor(tmpNeo, COLOR_BLACK);

   NeosRing.show();
   delay(InterPixelPadDelay);
}
```

Now, although this would be a perfectly acceptable solution, it still feels a bit 'graunchy,' if you know what I mean. What I usually do in this sort of case is perform thought experiments



(a) Using a single pixel

(b) Using multiple pixels

Fig.11. Two ways to reflect the position of the knob.

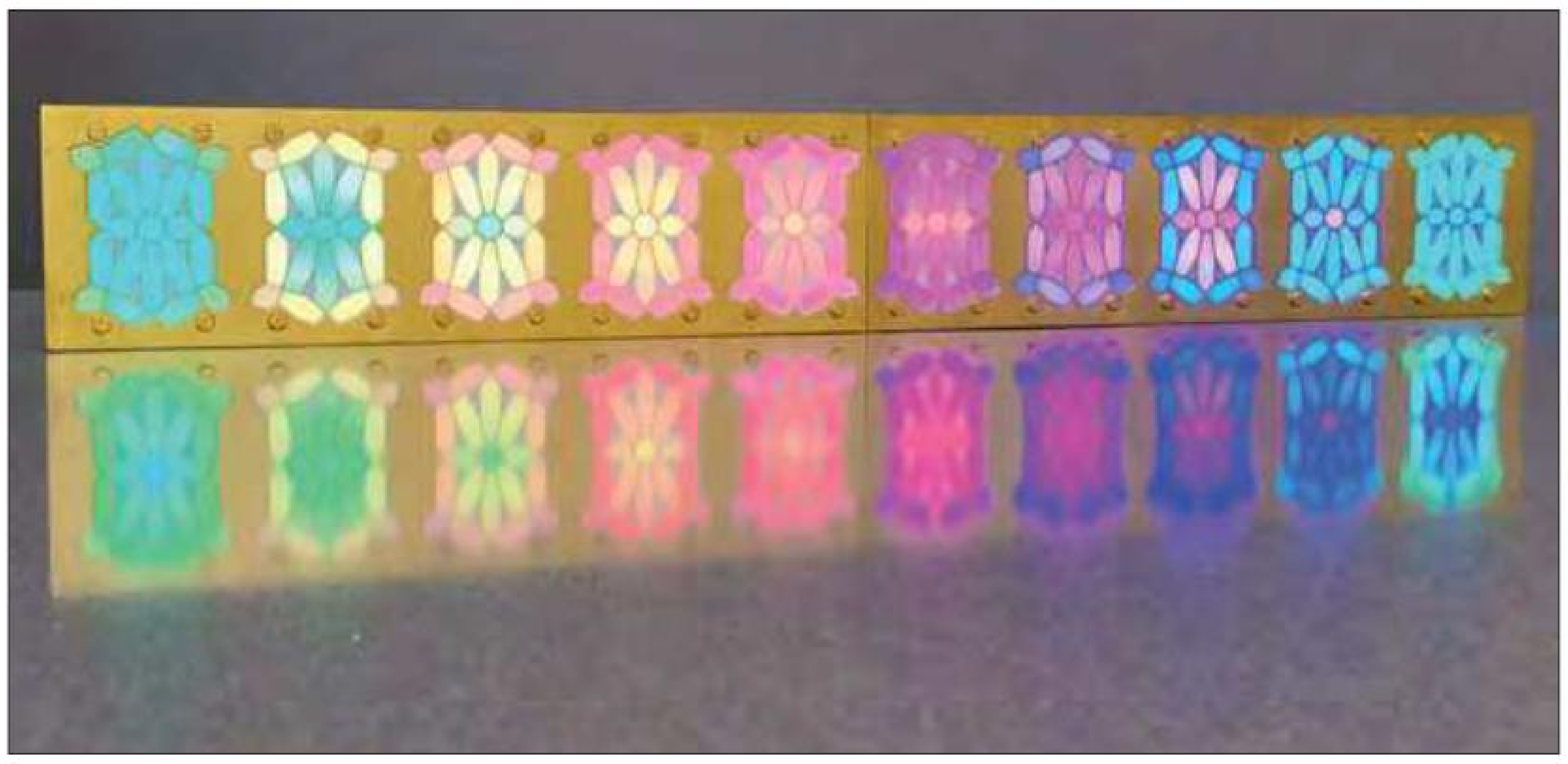


Fig.12. Steve Manley's array of ten 21-segment displays.

with pencil and paper. The first thing is to reduce the scope of the problem to a smaller number of pixels. Our original ring contains 16 pixels, which is an even number and a power of two (2^4 = 2^4). Given a choice, we want to come up with a generic solution that can work for any number of pixels, so we will play with a non-power of two. Furthermore, for this sort of thing, I prefer an odd number and a prime number on the basis that, if our solution works for this, it will work for anything.

So, purely for this thought experiment, let's assume that we have a 5-pixel ring. I start by drawing out a table (Fig.10). My first column lists what we've got (ie, what we know in the form of the information we have to hand), which is the number of the new pixel we're going to turn on. The next column lists what we want, which is the number of the old pixel we want to turn off.

I then dork around with various equations until I come up with a calculation that satisfies my needs. In this case, a simple calculation using the modulo operator `%` satisfies our requirements (this operator returns the integer remainder from an integer division). At this point, I transferred the equivalent over to my real-world code (file **CB-Jan21-06.txt**).

Now, our programs thus far have involved lighting our pixels in a clockwise fashion. Just for giggles and grins, let's suppose we want to modify our most recent sketch such that it lights the pixels in a widdershins (anticlockwise) pattern. Can you create your own table to determine the necessary calculation? – you can check out my solution in file **CB-Jan21-07.txt**

Turning the knob

Finally, for the moment, let's consider how we are going to translate the turning of the knob, which we read via our potentiometer, into the lighting of the pixels in our ring. Remember that we've decided not to access the bottom three pixels, which we will light magenta.

Let's assume that our On colour will be yellow and our Off colour will be red. Two main possibilities immediately spring to mind. The first is that we could light a single pixel to indicate the position of the pot (Fig.11a). The second is that we could light multiple pixels to indicate the position of the pot (Fig.11b).

You can take a look at my programs for both of these implementations in files **CB-Jan21-08.txt** and **CB-Jan21-09.txt**, respectively. You will observe that I've also added the gamma correction we introduced earlier into these final two sketches. Also, I just created a short video that shows all of the effects we've discussed here in action (**https://bit.ly/360O2hw**).

21-Segment Victorian Displays

Do you remember watching *Monty Python* when they said, 'And now for something completely different'? Well, hold onto your hat because here we go.

Way back in 1898 (123 years ago as I pen these words), someone called George Lafayette Mason filed a patent for a 21-segment display. These devices used 21 small incandescent bulbs (one per segment) to create letters, numbers, and symbols. A complicated electromechanical switch could be used to activate various groups of segments as required to represent the different characters.

The way I heard about this is that I have some friends (stop laughing; it's true) called Steve Manley and Paul Parry, where Paul is the owner of Bad Dog Designs (**https://bit.ly/2FTSBBk**). Paul and Steve ran across a group called Smartsockets (**https://bit.ly/2HOeFhw**), which was founded by Chris Barron and is moderated by Chris and John Smout.

In fact, it was John who came up with the idea of resurrecting these displays, but John and Chris are focused on their Smartsocket implementations in which each display has its own PIC microcontroller. By comparison, Paul, Steve, and myself prefer to drive all of our displays with a single Arduino-compatible microcontroller. As a result, the two sub-groups are heading in slightly different implementation directions, although they still communicate what they are doing with each other.

Steve has done a tremendous amount of work creating the circuit boards and 3D printed shells for these displays (Fig.12). Steve, Paul, and I each have ten of these displays, although I've not had the time to do anything with mine thus far. In fact, Steve has created a fantastic video that illustrates everything he's done in exquisite detail (**https://bit.ly/3oJwZcm**).

As we see in this image, Steve has used his 3D printer to make a pseudo-brass faceplate for his display. In my case, Paul introduced me to one of his friends, Kevin McIntosh. Kevin's company, the Laser Hut (**https://bit.ly/2Gf8qmg**) offers laser cutting and engraving services, and he's going to laser my pseudo-brass faceplates for me. I will, of course, be documenting all of this in excruciating detail in future columns.

Coming soon

As I've been known to say on occasion, 'Show me a flashing LED and I'll show you a man drooling' (hence the title of this miniseries). When we first started discussing the topic of flashing LEDs and drooling engineers (*PE*, March 2020), I really had no idea as to the myriad topics into which we were going to stick our snouts. In reality, of course, we've barely scratched the surface of this multifaceted topic, but I hope that the stuff we have covered has sparked your imagination and tempted you to toy with your own creations.

In the not-so-distant future, we're going to take a break from LEDs and turn our attention to other topics, but turn that frown upside down and turn it into a smile, because that time is not yet come. Next month, for example, we will be using our 12×12 array to implement a version of *Conway's Game of Life* – see: **https://bit.ly/pe-jan21-cgol**

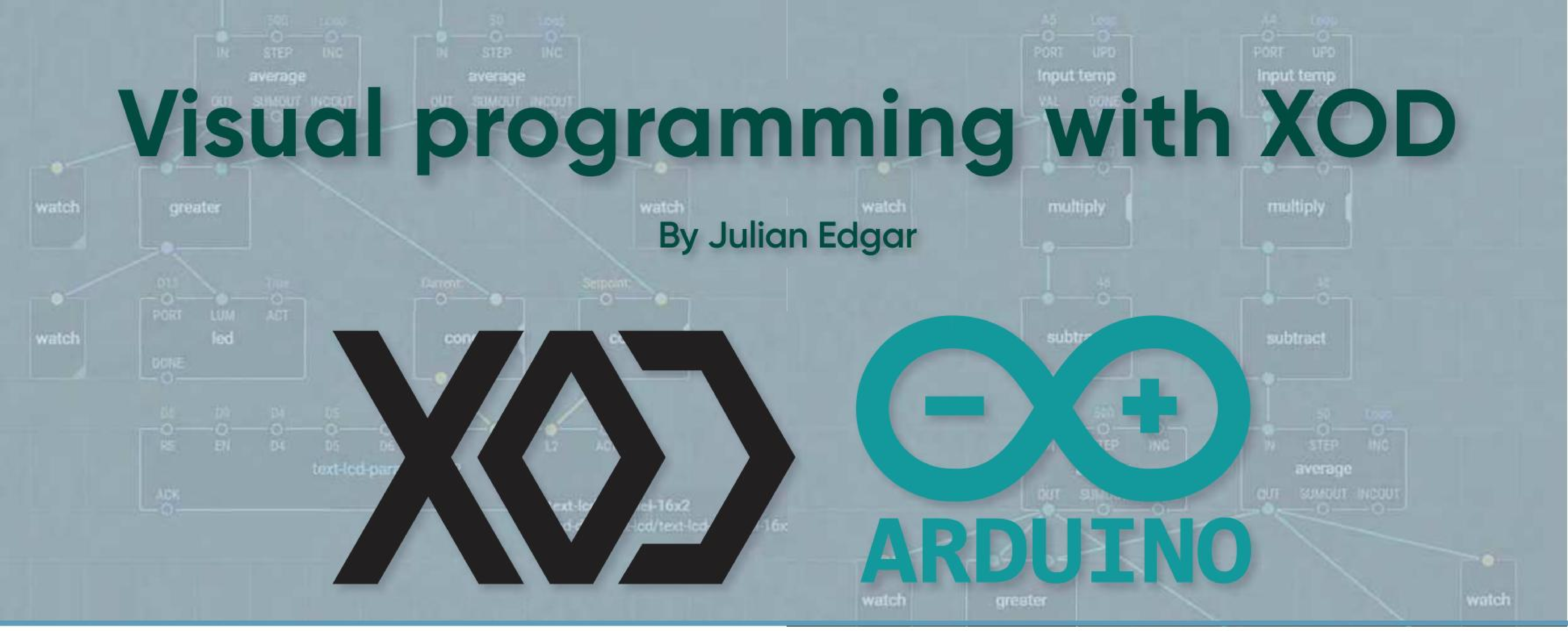
Until then, as always, I welcome your questions, comments, and suggestions.

Cool bean Max Maxfield (Hawaiian shirt, on the right) is emperor of all he surveys at **CliveMaxfield.com** – the go-to site for the latest and greatest in technological geekdom.

Comments or questions? Email Max at: **max@CliveMaxfield.com**

Visual programming with XOD

By Julian Edgar

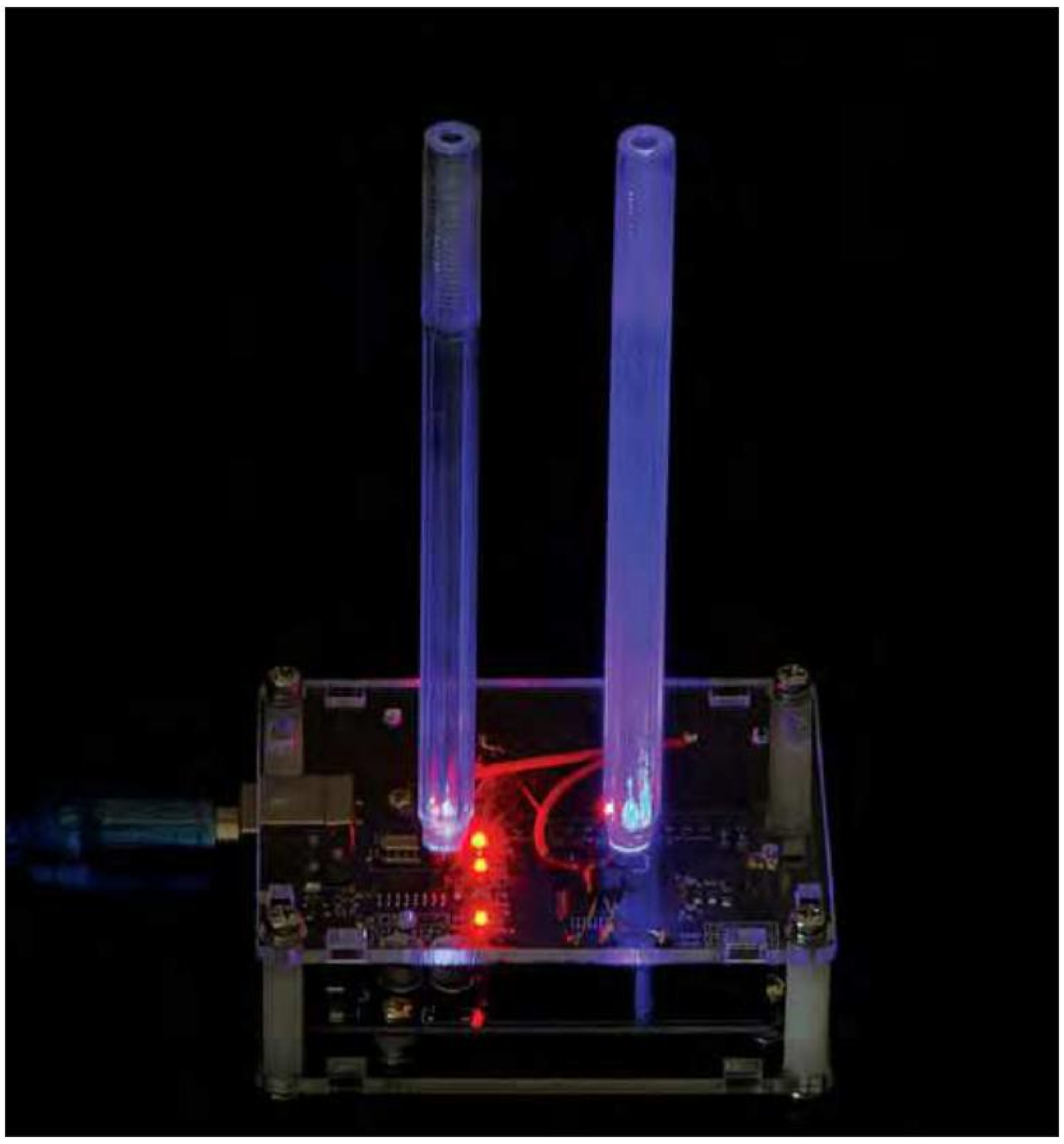


Light Column Thermometer

Picture the scene. You have some guests at your house and their eyes are attracted to a small device sitting on a shelf. The base is clear, and an electronics board can be seen, together with two projecting tubes, each pointing upwards. Sometimes the tubes fill with light, their entire columns illuminated. There seems to be a pattern in the way the tubes flash – but what is it?

First the two tubes light up together, one white and one blue. They flash for a few moments and then the white tube goes dark, just the blue tube continues to flash. Both then turn off, and the cycle starts again. It's obviously not random flashing, but what is it?

This time the guests count the flashes, and realise the white and blue tubes flash twice, and then the blue tube flashes three times. But no – this time around, the blue tube flashed four times! So, what is it? Realisation dawns – it's a thermometer, in this example reading 23°C and 24°C, respectively.



The *Light Column Thermometer* displays the temperature by using two illuminated plastic tubes. One tube shows 'tens' and the other 'units', with the number of flashes indicating the appropriate values.

This is a fun project – an indoor thermometer that displays the temperature in a unique way. It's cheap, very easy to construct and almost as easy to customise.

Cheap? The *Light Column Thermometer* uses an Arduino Uno board (clones now cost under £5 delivered), two clear plastic Uno enclosures (£2 each) and a handful of low-cost components. The light columns are recycled ballpoint pens illuminated by bright LEDs.

Easy to construct? Soldering will take you only moments and other than that, it's just a case of drilling some holes and screwing and gluing parts together.

Customisable? The program (sketch) for the Uno is written in XOD visual programming language. XOD (pronounced 'Zod') was introduced in the March 2020 issue of *PE* and is completely free to use. Unlike lines of code that often appear impenetrable, XOD is easy to understand and hence is easy to modify. If you wish, it will take you literally moments to alter the flash duration, flash fade speed – or even how often the cycle repeats. And you don't need to have white and blue flashing columns – just use whatever colour LEDs you prefer.

Building

The *Light Column Thermometer* uses the following parts, with the specific ones I used shown in brackets.

- Arduino Uno (eBay cheap clone – *see note below)
- Plastic Uno box (laser cut – I used two, eBay)
- Two pre-wired 5mm LEDs (blue and white, already fitted with dropping resistors – Banggood)
- LM35 DZ temperature sensor (eBay)
- Two discarded clear ballpoint pen barrels (salvaged)
- Assorted screws, nuts and spacers (in my parts box)

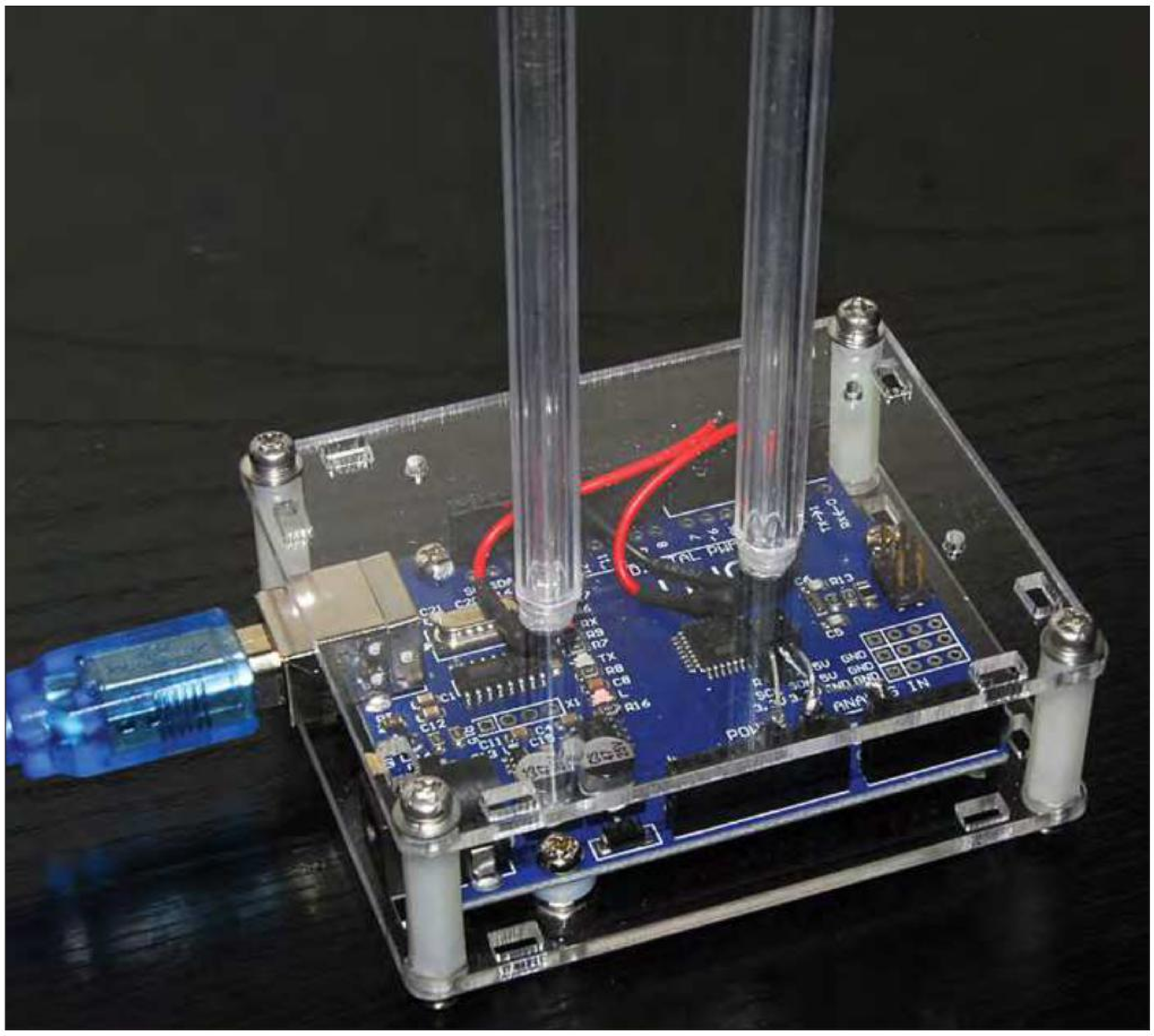
*Note: Some low-cost Uno modules do have one 'wrinkle'. Many use a non-standard USB communications chip, which if you are to communicate with it, needs a new PC driver. Some users report that Windows can find the driver by an automatic on-line search, but I had to download the driver from: **www.wch.cn/downloads/CH341SER_EXE.html**

If you are new to using Arduinos, perhaps initially stick with 'official' products – they don't need new drivers.

The enclosure uses the baseplates of two laser-cut boxes widely sold for use with the Uno. I used two baseplates (as opposed to the normal top and bottom parts) so that the top plate didn't have cut-outs in it. These cut-outs are provided to gain access to the pins, but I spaced the top and bottom of the enclosure further apart than usual (thus giving internal access to the pins) and so I didn't need the cut-outs. Of course, you can use just one box if you don't mind the slots in the top panel.

The top and bottom panels are spaced 20mm apart using plastic stand-offs, and the Uno is bolted to the baseplate via screws, plastic washers and nuts.

All the holes in the enclosure panels for mounting are drilled out to 3mm diameter, allowing the use of normal-sized spacers and screws. (As standard, these holes appear to be 2.5mm) The side and end plates of the enclosure are not used.



An Arduino Uno and two low-cost commercially available enclosures form the main components of the project. The upright columns are salvaged ballpoint pen bodies.

The light columns are, as described, salvaged ballpoint pen barrels. Pick a transparent design that has an interesting shape, preferably without writing on it. Different barrels will give different lighting effects; test the result by shining a 5mm LED down the end of the tube.

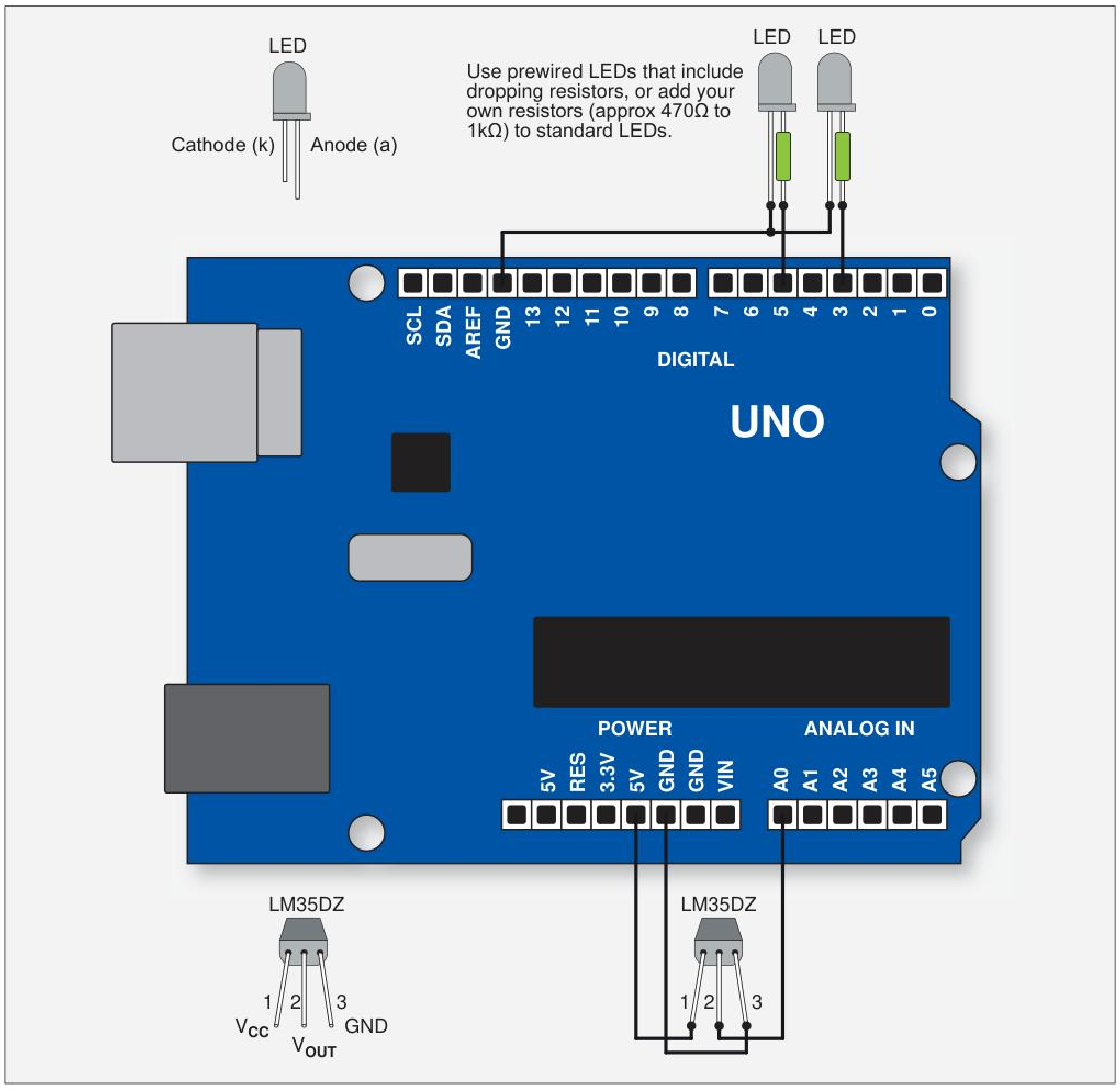


Fig.1. Connection diagram for the Arduino Uno, together with the pin-outs for the two LEDs and the LM35DZ temperature sensor. Its 5V supply is taken from the Arduino. Note: you can use LEDs prewired with dropper resistors, or ordinary LEDs and choose your own resistor.

The end of the pen barrels I used had a short length of exposed plastic thread. Two holes just undersize of this thread diameter were drilled in the top panel and then the plastic pen barrels could be screwed into the holes. A little cyanoacrylate glue ('superglue') was used to secure them into place. The pre-wired 5mm LEDs were glued into the ends of the tubes.

Wiring

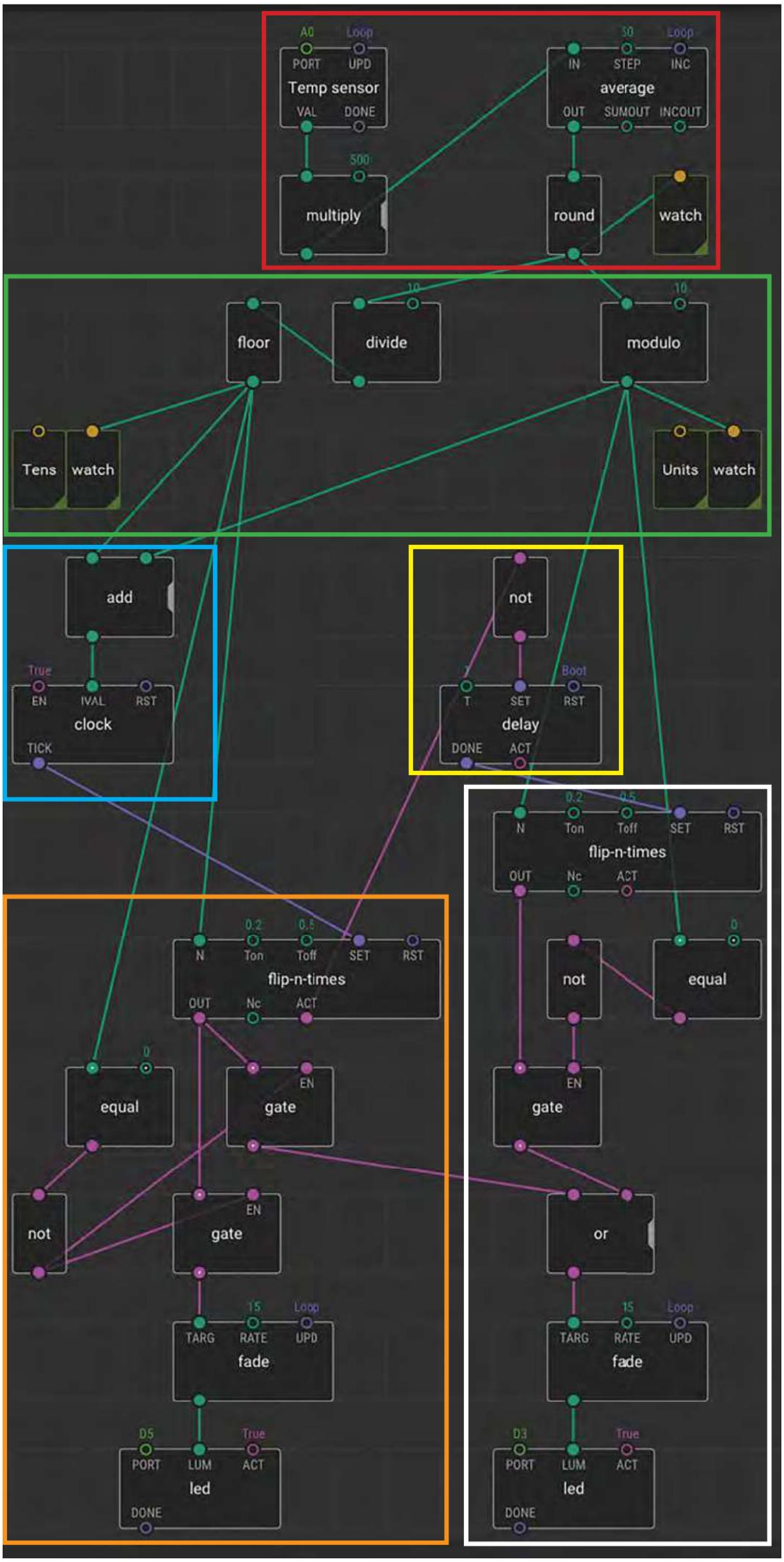
Refer to Fig.1 for the pinout of the LM35 temperature sensor. Port A0 is used for the signal, and the sensor's 5V and ground connections can be made close by on the Uno (all Uno ports are labelled). I used header pins and soldered the LM35 straight to these. Ensure that the signal wire cannot touch the ground connection.

The LEDs were wired between ground and ports D3 and D5, again using cut-off header pins. Remember the correct polarity for the LEDs – positives/anodes, to the ports; negatives/cathodes to ground.

Power to the board can be supplied via the DC socket (5-12V) or USB input.

Software

To upload the program (sketch) to the Uno you will have first needed to install XOD on your PC (see **https://xod.io/downloads/** – remember, it's free after you register). You will also need to download the *Light Column*


A0
Loop
PORT
UPD
Temp sensor
VAL
DONE
500
multiply
30
Loop
IN
STEP
INC
average
OUT
SUMOUT
INCOUT
round
watch
floor
10
divide
10
modulo
Tens
watch
Units
watch
add
True
EN
IVAL
RST
clock
TICK
not
1
Boot
T
SET
RST
delay
DONE
ACT
0.2
0.5
N
Ton
Toff
SET
RST
flip-n-times
OUT
Nc
ACT
0.2
0.5
N
Ton
Toff
SET
RST
flip-n-times
OUT
Nc
ACT
not
0
equal
0
equal
gate
EN
gate
EN
not
gate
EN
or
15
Loop
TARG
RATE
UPD
fade
15
Loop
TARG
RATE
UPD
fade
D5
True
PORT
LUM
ACT
led
DONE
D3
True
PORT
LUM
ACT
led
DONE

Thermometer sketch from the January 2021 page of the *PE* website. (Note that depending on whether you have used XOD previously, the software might prompt you to do some further downloading of extra libraries.)

Refer now to Fig.2. The beauty of XOD is that's it very easy to understand. In the red box (top of diagram) we have the input from the temperature sensor, constantly read through Analog Port A0. This value is multiplied by 500, averaged and then rounded. (The 'live' temperature is shown in the green 'watch' node, that operates when the sketch is uploaded to the Uno in 'debug' mode.

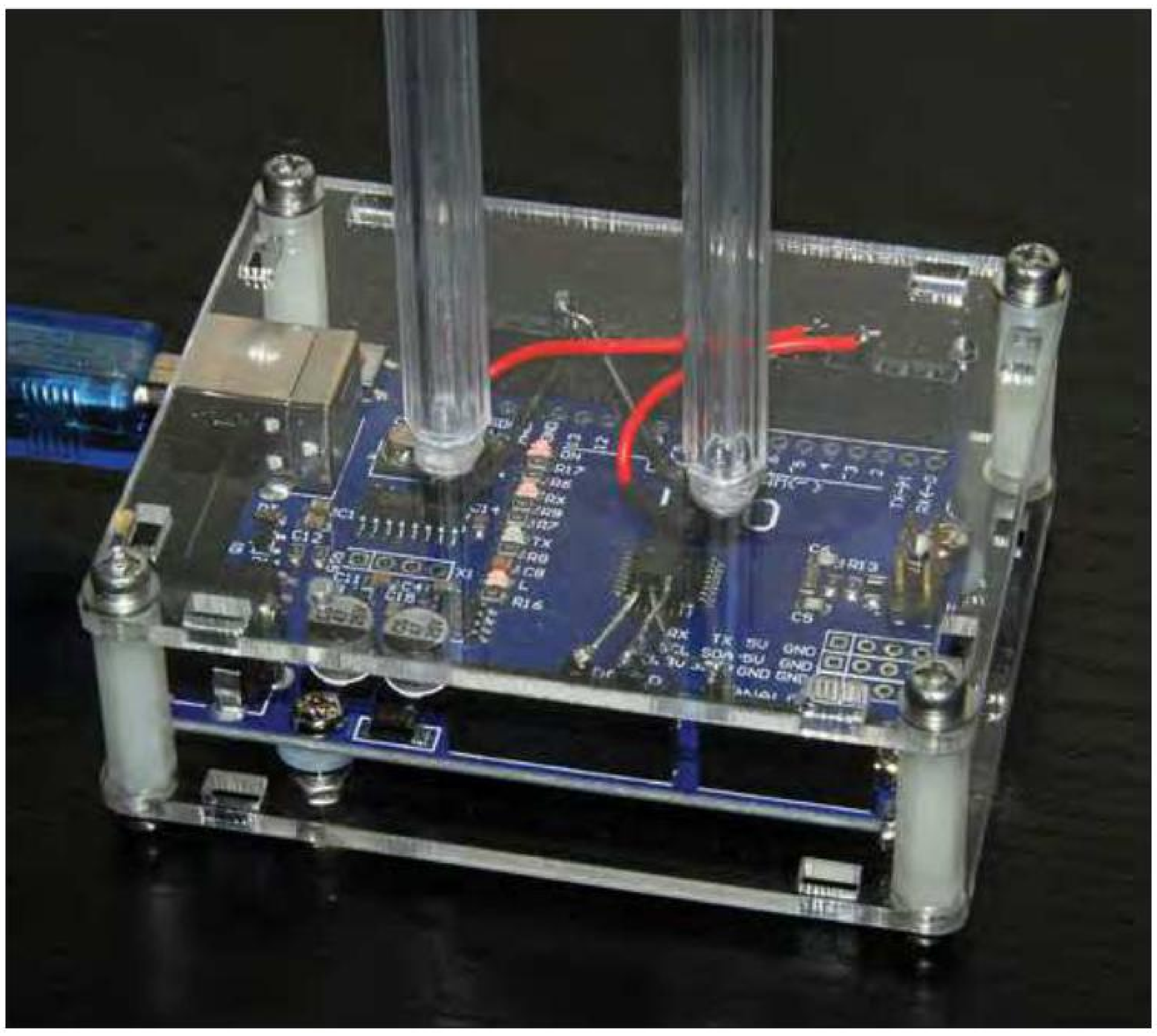
We now need to extract from this number the 'tens' and 'units' – see the green box. Dividing the value by 10 and then using a 'floor' node does this for the 'tens'. Now, what about the 'units'? The 'modulo' node does this by calculating the remainder of (again) dividing our temperature value by ten. Two 'watch' nodes allow us to see these outputs live.

Let's do the white box next – the flashing shows the number of 'units'. Our 'units' number is fed to a 'flip-n-times' node. This node flashes the LED output the required number of times, and also sets the flash rate and duty cycle – in this case, 0.2 seconds 'on' and 0.5 seconds 'off'. The 'gate', 'not' and 'equal' nodes then prevent an output if the 'units' number is zero. (Otherwise, the 'flip-n-times' node outputs one flash, even with a 0 input.) We then feed the output through an 'or' node (more on this in a moment) and then through a 'fade' node. The fade node gives a gradual (although still pretty fast) rise and fall in LED brightness with each flash.

Now, what about the 'tens'? The tricky part here is twofold: first, the 'units' column needs to flash at the same time as the 'tens' column when 'tens' are being shown, and second, the 'units' can't start to flash until the 'tens' have finished their sequence.

Flashing the 'units' LED when the 'tens' LED is flashing is achieved by the 'or' node. But what about not starting the 'units' until the 'tens' are done? This is done

Fig.2. The Arduino sketch for the *Light Column Thermometer* is written in XOD visual programming language. The sketch is fully explained in the main text, but in brief, the red box shows the temperature input nodes, the yellow box extracts the 'tens' and 'units' from the reading, the white box flashes the LED for the 'units' reading, the brown box flashes the LED for the 'tens' reading, the yellow box starts the 'units' only after the 'tens' have finished, and the blue box sets the cycle time.



Spacers are used to hold the two parts of the enclosure 20mm apart. This gives enough room for the wiring of the temperature sensor and two LEDs to be placed within the enclosure.

by the contents of the yellow box – that registers when the 'tens' have finished, adds a delay of one second, and then starts the 'units' flashing.

The brown box comprises the logic for the 'tens' flasher; it uses the same approach as for the 'units'.

The blue box starts the cycle. It does this by adding the 'tens' and 'units' numbers together to gain the total required number of flashes. In this example that is six (two 'tens' flashes and four 'units' flashes). Each flash takes 0.7 seconds (0.5 off and 0.2 on), giving a total of 4.2 seconds. Add to that the one-second delay between the 'tens' finishing flashing and the 'units' starting, and we have 5.2 seconds for the sequence. Since the clock resets at (in this example) six seconds, we have a 0.8 second delay at the end of the cycle before it starts again. Taking this approach means the cycle time adapts to the required number of flashes.

As with all XOD sketches, this one is easily customised. For example, if you wanted the cycle time to be longer, you could put another 'add' node in above the 'clock' and add a further (say) five seconds to the cycle time.

Note that as configured, the thermometer cannot show negative temperatures, nor degrees Fahrenheit. (Both of these are quite possible, though. For example, XOD even has available a direct units conversion node for Celsius/Fahrenheit – you might want to try adding it to the sketch.)

Conclusion

I find the *Light Column Thermometer* fascinating to watch. It can be seen from across the other side of the room, and reading it is oddly relaxing – perhaps because you have to pause and watch, rather than just glance and run. It's also an ideal project with which to discover the fascinating world of XOD.

XOD files

The XOD file discussed in this article can be downloaded from the January 2021 page of the *PE* website.

See *PE* March 2020 for an introduction to using XOD.

ELECTRONICS TEACH-IN 8
INTRODUCING THE ARDUINO
£8.99
FREE CD-ROM

Teach-In 8 CD-ROM
Exploring the Arduino

This CD-ROM version of the exciting and popular *Teach-In 8* series has been designed for electronics enthusiasts who want to get to grips with the inexpensive, immensely popular Arduino microcontroller, as well as coding enthusiasts who want to explore hardware and interfacing. *Teach-In 8* provides a one-stop source of ideas and practical information.

The Arduino offers a remarkably effective platform for developing a huge variety of projects; from operating a set of Christmas tree lights to remotely controlling a robotic vehicle wirelessly or via the Internet. *Teach-In 8* is based around a series of practical projects with plenty of information for customisation. The projects can be combined together in many different ways in order to build more complex systems that can be used to solve a wide variety of home automation and environmental monitoring problems. The series includes topics such as RF technology, wireless networking and remote web access.

PRICE £8.99
Includes P&P to UK if ordered direct from us

PLUS: PICs and the PICkit 3 – A beginners guide
The CD-ROM also includes a bonus – an extra 12-part series based around the popular PIC microcontroller, explaining how to build PIC-based systems.

SOFTWARE
The CD-ROM contains the software for both the *Teach-In 8* and PICkit 3 series.

ORDER YOUR COPY TODAY!
JUST CALL 01202 880299 OR VISIT www.electronpublishing.com

CLASSIFIED ADVERTISING

Practical Electronics

If you want your advertisements to be seen by the largest readership at the most economical price then our classified page offers excellent value. The rate for semi-display space is £10 (+VAT) per centimetre high, with a minimum height of 2·5cm. All semi-display adverts have a width of 5.5cm. The prepaid rate for classified adverts is 40p (+VAT) per word (minimum 12 words).

Cheques are made payable to 'Practical Electronics'. **VAT must be added**. Advertisements with remittance should be sent to: Practical Electronics, 113 Lynwood Drive, Wimborne, Dorset, BH21 1UU.

Tel 07973518682 Email: pe@electronpublishing.com

For rates and further information on display and classified advertising please contact our Advertisement Manager, Matt Pulzer – see below.

Practical Electronics reaches more UK readers than any other UK monthly hobby electronics magazine. Our sales figures prove it.

We have been the leading monthly magazine in this market for the last twenty-seven years.

BOWOOD ELECTRONICS LTD
Suppliers of Electronic Components
www.bowood-electronics.co.uk
Unit 10, Boythorpe Business Park, Dock Walk, Chesterfield, Derbyshire S40 2QR. Sales: 01246 200 222
Send large letter stamp for Catalogue

Electrical Industries Charity (EIC)
We help people working in the electrical, electronics and energy community as well as their family members and retirees.
We use workplace programmes that give the industry access to financial grants and a comprehensive range of free and confidential services.
www.electricalcharity.org
electrical industries charity

COAST ELECTRONICS
BREAKOUTS-COMPONENTS-CONTRACT DESIGN-3D PRINTER PARTS-MUSICAL-MICROCONTROLLERS
WWW.COASTELECTRONICS.CO.UK

Andrew Kenny – Qualified Patent Agent
EPO UKIPO USPTO
Circuits Electric Machinery Mechatronics
Web: www.akennypatentm.com
Email: Enquiries@akennypatentm.com
Tel: 0789 606 9725

MISCELLANEOUS

VALVES AND ALLIED COMPONENTS IN STOCK. Phone for free list. Valves, books and magazines wanted. Geoff Davies (Radio), tel. 01788 574774.

PIC DEVELOPMENT KITS, DTMF kits and modules, CTCSS Encoder and Decoder/Display kits.
Visit www.cstech.co.uk

ADVERTISING INDEX

CRICKLEWOOD ELECTRONICS 57
ESR ELECTRONIC COMPONENTS........................ 57
HAMMOND ELECTRONICS Ltd 8
JPG ELECTRONICS 72
MICROCHIP................................... Cover (ii)
PEAK ELECTRONIC DESIGN...................... Cover (iv)
POLABS D.O.O....................................... 49
QUASAR ELECTRONICS 3
SILICON CHIP 5
STEWART OF READING................................ 9
TAG-CONNECT 57

Advertisement offices
Matt Pulzer
Electron Publishing Ltd
1 Buckingham Road
Brighton
East Sussex BN1 3RA
Tel 07973 518682
Email pe@electronpublishing.com
Web www.electronpublishing.com

For editorial contact details see page 7.

Practical Electronics PCB SERVICE

PROJECT	CODE	PRICE
JANUARY 2021		
Nutube Valve Preamplifier	01112191	£12.95
Arduino DCC Controller	09207181	£10.95
DECEMBER 2020		
Pseudo-Random Sequence Generator	16106191	£7.95
Clever Charger	14107191	£11.95
PE Theremin Amplifier	AO-1220-01	£7.95
NOVEMBER 2020		
LED Christmas Tree (1 off)	16107181-1	£6.95
LED Christmas Tree (4 off)	16107181-2	£14.95
LED Christmas Tree (12 off)	16107181-3	£24.95
LED Christmas Tree (20 off)	16107181-4	£34.95
USB/SPI Interface Board	16107182	£5.95
45V/8A Power Supply PCB *plus* acrylic spacer	18111181	£14.95
45V/8A Power Supply front panel five-way display bezel	18111181-BZ	£3.95
Five-way LCD Panel Meter/Display	18111182	£7.95
OCTOBER 2020		
Digital Audio Millivoltmeter	04108191	£8.95
Precision Signal Amplifier	04107191	£6.95
SEPTEMBER 2020		
PE Theremin PSU	AO-0920-01	£5.95
PE Theremin PSU transformer	AO-0920-02	£7.95
Micromite Explore-28	07108191	£5.95
Ultrabrite LED Driver	16109191	£5.95
AUGUST 2020		
Micromite LCD BackPack V3	07106191	£7.95
Steering Wheel Audio Button to Infrared Adaptor	05105191	£7.95
JULY 2020		
AM/FM/CW Scanning HF/VHF RF Signal Generator	04106191	£11.95
Speech Synthesiser with the Raspberry Pi Zero	01106191	£5.95
PE Mini-organ PCB	AO-0720-01	£14.95
PE Mini-organ selected parts	AO-0720-02	£8.95
High-current Solid-state 12V Battery Isolator – control	05106191	£6.95
High-current Solid-state 12V Battery Isolator FET (2oz)	05106192	£9.95
JUNE 2020		
Arduino breakout board – 3.5-inch LCD Display	24111181	£6.95
Six-input Audio Selector main board	01110191	10.95
Six-input Audio Selector switch panel board	01110192	
MAY 2020		
Ultra-low-distortion Preamplifier Input Selector	01111112	11.25
Ultra-low-distortion Preamplifier pushbutton Input Selector	01111113	
Universal Regulator	18103111	7.95
433MHz Wireless Data Repeater	15004191	8.50
Bridge-mode Adaptor for Amplifier	01105191	7.95
iCEstick VGA Terminal	02103191	4.95
Analogue noise with tilt control	AO-0520-01	7.95
Audio Spectrum Analyser	PM-0520-01	8.95
APRIL 2020		
Flip-dot Display black coil board	19111181	£14.95
Flip-dot Display black pixels	19111182	
Flip-dot Display black frame	19111183	
Flip-dot Display green driver board	19111184	
MARCH 2020		
Diode Curve Plotter	04112181	£10.95
Steam Train Whistle / Diesel Horn Sound Generator	09106181	£8.50
Universal Passive Crossover (one off)	UPC0320	£12.50

PROJECT	CODE	PRICE
FEBRUARY 2020		
Motion-Sensing 12V Power Switch	05102191	£5.95
USB Keyboard / Mouse Adaptor	24311181	£8.50
DSP Active Crossover (ADC)	01106191	£29.95
DSP Active Crossover (DAC) ×2	01106192	
DSP Active Crossover (CPU)	01106193	
DSP Active Crossover (Power/routing)	01106194	
DSP Active Crossover (Front panel)	01106195	
DSP Active Crossover (LCD)	01106196	
JANUARY 2020		
Isolated Serial Link	24107181	£8.50
DECEMBER 2019		
Extremely Sensitive Magnetometer	04101011	£16.75
Four-channel High-current DC Fan and Pump Controller	05108181	£8.75
Useless Box	08111181	£11.50
NOVEMBER 2019		
Tinnitus & Insomnia Killer (Jaycar case – see text)	01110181	£8.75
Tinnitus & Insomnia Killer (Altronics case – see text)	01110182	£8.75
OCTOBER 2019		
Programmable GPS-synced Frequency Reference	04107181	£11.50
Digital Command Control Programmer for Decoders	09107181	£8.75
Opto-isolated Mains Relay (main board)	10107181	£11.50
Opto-isolated Mains Relay (2 × terminal extension board)	10107182	
AUGUST 2019		
Brainwave Monitor	25108181	£12.90
Super Digital Sound Effects Module	01107181	£5.60
Watchdog Alarm	03107181	£8.00
PE Theremin (three boards: pitch, volume, VCA)	PETX0819	£19.50
PE Theremin component pack (see p.56, August 2019)	PETY0819	£15.00
JULY 2019		
Full-wave 10A Universal Motor Speed Controller	10102181	£12.90
Recurring Event Reminder	19107181	£8.00
Temperature Switch Mk2	05105181	£10.45
JUNE 2019		
Arduino-based LC Meter	04106181	£8.00
USB Flexitimer	19106181	£10.45
MAY 2019		
2× 12V Battery Balancer	14106181	£5.60
Deluxe Frequency Switch	05104181	£10.45
USB Port Protector	07105181	£5.60
APRIL 2019		
Heater Controller	10104181	£14.00
MARCH 2019		
10-LED Bargraph Main Board	04101181	£11.25
+Processing Board	04101182	£8.60
FEBRUARY 2019		
1.5kW Induction Motor Speed Controller	10105122	£35.00
NOVEMBER 2018		
Super-7 AM Radio Receiver	06111171	£27.50
OCTOBER 2018		
6GHz+ Touchscreen Frequency Counter	04110171	£12.88
Two 230VAC MainsTimers	10108161 10108162	£12.88

PCBs for most recent *PE*/*EPE* constructional projects are available. From the July 2013 issue onwards, PCBs with eight-digit codes have silk screen overlays and, where applicable, are double-sided, have plated-through holes, and solder mask. They are similar to photos in the project articles. Earlier PCBs are likely to be more basic and may not include silk screen overlay, be single-sided, lack plated-through holes and solder mask.

Always check price and availability in the latest issue or online. A large number of older boards are listed for ordering on our website.

In most cases we do not supply kits or components for our projects. For older projects it is important to check the availability of all components before purchasing PCBs.

Back issues of articles are available – see Back Issues page for details.

Double-sided | plated-through holes | solder mask

PROJECT	CODE	PRICE
SEPTEMBER 2018		
3-Way Active Crossover	01108171	£22.60
Ultra-low-voltage Mini LED Flasher	16110161	£5.60
AUGUST 2018		
Universal Temperature Alarm	03105161	£7.05
Power Supply For Battery-Operated Valve Radios	18108171	£27.50
	18108172	
	18108173	
	18108174	
JULY 2018		
Touchscreen Appliance Energy Meter – Part 1	04116061	£17.75
Automotive Sensor Modifier	05111161	£12.88
JUNE 2018		
High Performance 10-Octave Stereo Graphic Equaliser	01105171	£15.30
MAY 2018		
High Performance RF Prescaler	04112162	£10.45
Micromite BackPack V2	07104171	£10.45
Microbridge	24104171	£5.60
APRIL 2018		
Spring Reverberation Unit	01104171	£15.30
DDS Sig Gen Lid	Black	£8.05
DDS Sig Gen Lid	Blue	£7.05
DDS Sig Gen Lid	Clear	£8.05
MARCH 2018		
Stationmaster Main Board	09103171	£17.75
+ Controller Board	09103172	
SC200 Amplifier Module – Power Supply	01109111	£16.45
FEBRUARY 2018		
GPS-Synchronised Analogue Clock Driver	04202171	£12.88
High-Power DC Motor Speed Controller – Part 2		
+ Control Board	11112161	£12.88
+ Power Board	11112162	£15.30
JANUARY 2018		
High-Power DC Motor Speed Controller – Part 1	11112161	£12.88
Build the SC200 Amplifier Module	01108161	£12.88
DECEMBER 2017		
Precision Voltage and Current Reference – Part 2	04110161	£15.35
NOVEMBER 2017		
50A Battery Charger Controller	11111161	£12.88
Micropower LED Flasher (45 × 47mm)	16109161	£8.00
(36 × 13mm)	16109162	£5.60
Phono Input Converter	01111161	£8.00
SEPTEMBER 2017		
Compact 8-Digit Frequency Meter	04105161	£12.88
AUGUST 2017		
Micromite-Based Touch-screen Boat Computer GPS	07102122	£10.45
Fridge/Freezer Alarm	03104161	£8.00
JULY 2017		
Micromite-Based Super Clock	07102122	£10.45
Brownout Protector for Induction Motors	10107161	£12.90
JUNE 2017		
Ultrasonic Garage Parking Assistant	07102122	£10.45
Hotel Safe Alarm	03106161	£8.00
100dB Stereo LED Audio Level/VU Meter	01104161	£17.75

PROJECT	CODE	PRICE
MAY 2017		
The Micromite LCD BackPack	07102122	£11.25
Precision 230V/115V 50/60Hz Turntable Driver	04104161	£19.35
APRIL 2017		
Microwave Leakage Detector	04103161	£8.00
Arduino Multifunctional 24-bit Measuring Shield	04116011	£17.75
+ RF Head Board	04116012	
Battery Pack Cell Balancer	11111151	£9.00
MARCH 2017		
Speech Timer for Contests & Debates	19111151	£16.42
FEBRUARY 2017		
Solar MPPT Charger/Lighting Controller	16101161	£17.75
Turntable LED Strobe	04101161	£7.60
JANUARY 2017		
High-performance Stereo Valve Preamplifier	01101161	£17.75
High Visibility 6-Digit LED Clock	19110151	£16.42

For the many pre-2017 PCBs that we stock please see the PE website: www.electronpublishing.com

PE/EPE PCB SERVICE

Order Code	Project	Quantity	Price
.....	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....

Name ....

Address ....

....

Tel ....

Email ....

I enclose payment of £.............. (cheque/PO in **£ sterling only**)

payable to: **Practical Electronics**

Card No ....

Valid From ....Expiry Date ....

Card Security No ....

You can also order PCBs by phone, email or via the shop on our website: www.electronpublishing.com

No need to cut your issue – a copy of this form is just as good!

All prices include VAT and UK p&p. Add £4 per *project* for post to Europe; £5 per *project* outside Europe.

Orders and payment should be sent to:

Practical Electronics, Electron Publishing Ltd
113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU
Tel 01202 880299 Email: shop@electronpublishing.com
On-line Shop: www.epemag.com

Cheques should be made payable to 'Practical Electronics' **(Payment in £ sterling only)**.

NOTE: Most boards are in stock and sent within seven days of receipt of order, please allow up to 28 days delivery if we need to restock.

DIRECT BOOK SERVICE

Teach-In 2017

Introducing the BBC micro:bit

IN FULL COLOUR
For Beginners of Any Age
Python Coding on the BBC micro:bit
Jim Gatenby

PYTHON CODING ON THE BBC MICRO:BIT

Jim Gatenby

Python is the leading programming language, easy to learn and widely used by professional programmers. This book uses MicroPython, a version of Python adapted for the BBC Micro:bit.

Among the many topics covered are: main features of the BBC micro:bit including a simulation in a web browser screen; various levels of programming languages; Mu Editor for writing, saving and retrieving programs, with sample programs and practice exercises; REPL, an interactive program for quickly testing lines of code; scrolling messages, creating and animating images on the micro:bit's LEDs; playing and creating music, sounds and synthesized speech; using the on-board accelerometer to detect movement of the micro:bit on three axes; glossary of computing terms.

This book is written using plain English, avoids technical jargon wherever possible and covers many of the coding instructions and methods which are common to most programming languages. It should be helpful to beginners of any age, whether planning a career in computing or writing code as an enjoyable hobby.

118 Pages **Order code PYTH MBIT** **£7.99**

IN FULL COLOUR
For Beginners of Any Age
Getting Started with the BBC micro:bit
Mike Tooley

GETTING STARTED WITH THE BBC MICRO:BIT

Mike Tooley

Not just an educational resource for teaching youngsters coding, the BBC micro:bit is a tiny low cost, low-profile ARM-based single-board computer. The board measures 43mm × 52mm but despite its diminutive footprint it has all the features of a fully fledged microcontroller together with a simple LED matrix display, two buttons, an accelerometer and a magnetometer.

Mike Tooley's book will show you how the micro:bit can be used in a wide range of applications from simple domestic gadgets to more complex control systems such as those used for lighting, central heating and security applications. Using Microsoft Code Blocks, the book provides a progressive introduction to coding as well as interfacing with sensors and transducers.

Each chapter concludes with a simple practical project that puts into practice what the reader has learned. The featured projects include an electronic direction finder, frost alarm, reaction tester, battery checker, thermostatic controller and a passive infrared (PIR) security alarm.

No previous coding experience is assumed, making this book ideal for complete beginners as well as those with some previous knowledge. Self-test questions are provided at the end of each chapter, together with answers at the end of the book. So whatever your starting point, this book will take you further along the road to developing and coding your own real-world applications.

108 Pages **Order code BBC MBIT** **£7.99**

THEORY AND REFERENCE

PRACTICAL ELECTRONICS HANDBOOK – 6th Ed
Ian Sinclair
440 pages **Order code NE21** **£33.99**

STARTING ELECTRONICS – 4th Ed
Keith Brindley
296 pages **Order code ELSEV100** **£18.99**

ELECTRONIC CIRCUITS – FUNDAMENTALS & APPLICATIONS – Updated version
Mike Tooley
400 pages **Order code TF43** **£32.99**

FUNDAMENTAL ELECTRICAL AND ELECTRONIC PRINCIPLES – 3rd Ed
C.R. Robertson
368 pages **Order code TF47** **£21.99**

A BEGINNER'S GUIDE TO TTL DIGITAL ICs
Robert Penfold
142 pages **OUT OF PRINT BP332** **£5.45**

UNDERSTANDING ELECTRONIC CONTROL SYSTEMS
Owen Bishop
228 pages **Order code NE35** **£36.99**

The books listed here have been selected by the *Practical Electronics* editorial staff as being of special interest to everyone involved in electronics and computing. They are supplied by mail order direct to your door.

FOR A FULL DESCRIPTION OF THESE BOOKS AND CD-ROMS SEE THE SHOP ON OUR WEBSITE

www.electronpublishing.com

All prices include UK postage

MICROPROCESSORS

INTERFACING PIC MICROCONTROLLERS – 2nd Ed
Martin Bates
298 pages **Order code NE48** **£34.99**

PROGRAMMING 16-BIT PIC MICROCONTROLLERS IN C – LEARNING TO FLY THE PIC24
Lucio Di Jasio (Application Segments Manager, Microchip, USA)
496 pages + CD-ROM **Order code NE45** **£38.00**

INTRODUCTION TO MICROPROCESSORS AND MICROCONTROLLERS – 2nd Ed
John Crisp
222 pages **Order code NE31** **£29.99**

THE PIC MICROCONTROLLER YOUR PERSONAL INTRODUCTORY COURSE – 3rd Ed
John Morton
270 pages **Order code NE36** **£25.00**

PIC IN PRACTICE – 2nd Ed
David W. Smith
308 pages **Order code NE39** **£24.99**

MICROCONTROLLER COOKBOOK
Mike James
240 pages **Order code NE26** **£36.99**

BOOK ORDERING DETAILS

All prices include UK postage.

For postage, add £3 per book to Europe, £4 for rest of the world per book.

CD-ROM prices include VAT and/or postage to anywhere in the world.

Send a cheque, (£ sterling only) made payable to: **Practical Electronics** or credit card details (Visa or Mastercard) to:

Electron Publishing Limited,
113 Lynwood Drive, Wimborne, Dorset BH21 1UU

Books are normally sent within seven days of receipt of order.

Please check price (see latest issue of Practical Electronics or website) before ordering from old lists.

For a full description of these books please see the shop on our website.
Tel: 01202 880299 – Email: shop@electronpublishing.com

Order from our online shop at: www.electronpublishing.com

ARDUINO

ARDUINO FOR DUMMIES

John Nussey

Arduino is no ordinary circuit board. Whether you're an artist, a designer, a programmer, or a hobbyist, Arduino lets you learn about and play with electronics. You'll discover how to build a variety of circuits that can sense or control real-world objects, prototype your own product, and even create interactive artwork. This handy guide is exactly what you need to build your own Arduino project – what you make is up to you!

- **Learn by doing** – start building circuits and programming your Arduino with a few easy examples – right away!
- **Easy does it** – work through Arduino sketches line by line, and learn how they work and how to write your own.
- **Solder on!** – don't know a soldering iron from a curling iron? No problem! You'll learn the basics and be prototyping in no time.
- **Kitted out** – discover new and interesting hardware to turn your Arduino into anything from a mobile phone to a Geiger counter.
- **Become an Arduino savant** – find out about functions, arrays, libraries, shields and other tools that let you take your Arduino project to the next level
- **Get social** – teach your Arduino to communicate with software running on a computer to link the physical world with the virtual world

438 Pages **Order code ARDDUM01** £19.99

EXPLORING ARDUINO

Jeremy Blum

Arduino can take you anywhere. This book is the roadmap.

Exploring Arduino shows how to use the world's most popular microcontroller to create cool, practical, artistic and educational projects. Through lessons in electrical engineering, programming and human-computer interaction, this book walks you through specific, increasingly complex projects, all the while providing best practices that you can apply to your own projects once you've mastered these. You'll acquire valuable skills – and have a whole lot of fun.

- **Explore** the features of commonly used Arduino boards
- **Use** Arduino to control simple tasks or complex electronics
- **Learn** principles of system design, programming and electrical engineering
- **Discover** code snippets, best practices and system schematics you can apply to your original projects
- **Master** skills you can use for engineering endeavours in other fields and with different platforms

357 Pages **Order code EXPARD01** £26.99

COMPUTING AND ROBOTICS

NEWNES INTERFACING COMPANION
Tony Fischer-Cripps
295 pages **Order code NE38** £41.00

HOW TO BUILD A COMPUTER MADE EASY
Robert Penfold
120 pages **Order code BP707** £8.49

EASY PC CASE MODDING
Robert Penfold
192 pages + CD-ROM **Order code BP542** £8.99

FREE DOWNLOADS TO PEP-UP AND PROTECT YOUR PC
Robert Penfold
128 pages **Order code BP722** £7.99

WINDOWS XP EXPLAINED
N. Kantaris and P.R.M. Oliver
264 pages **Order code BP514** £7.99

THE INTERNET – TWEAKS, TIPS AND TRICKS
Robert Penfold
128 pages **Order code BP721** £7.99

eBAY – TWEAKS, TIPS AND TRICKS
Robert Penfold
128 pages **Order code BP716** £7.50

AN INTRODUCTION TO eBAY FOR THE OLDER GENERATION
Cherry Nixon
120 pages **Order code BP709** £8.49

HOW TO FIX YOUR PC PROBLEMS
Robert Penfold
128 pages **Order code BP705** £8.49

AN INTRODUCTION TO WINDOWS VISTA
P.R.M. Oliver and N. Kantarris
120 pages **Order code BP703** £8.49

WINDOWS 8.1 EXPLAINED
180 Pages **Order code BP747** £10.99

AN INTRODUCTION TO THE NEXUS 7
118 Pages **Order code BP744** £8.99

COMPUTING FOR THE OLDER GENERATION
Jim Gatenby
308 pages **Order code BP601** £8.99

ANDROIDS, ROBOTS AND ANIMATRONS
Second Edition – John Iovine
224 pages **Order code MGH1** £16.99

ROBOT BUILDERS COOKBOOK
Owen Bishop
366 pages **Order code NE46** £26.00

INTRODUCING ROBOTICS WITH LEGO MINDSTORMS
Robert Penfold
288 pages + **Order code BP901** £14.99

MORE ADVANCED ROBOTICS WITH LEGO MINDSTORMS
Robert Penfold
298 pages **Order code BP902** £14.99

HOW TO FIX YOUR PC PROBLEMS
Robert Penfold
128 pages **Order code BP705** £8.49

WINDOWS 7 – TWEAKS, TIPS AND TRICKS
Andrew Edney
120 pages **Order code BP708** £8.49

GETTING STARTED IN COMPUTING FOR THE OLDER GENERATION
Jim Gatenby
120 pages **Order code BP704** £8.49

WINDOWS 8.1 EXPLAINED
Noel Kantaris
180 Pages **Order code BP747** £10.99

COMPUTING WITH A LAPTOP FOR THE OLDER GENERATION
Robert Penfold
120 pages **Order code BP702** £8.49

AN INTRODUCTION TO EXCEL SPREADSHEETS
Jim Gatenby
18 pages **Order code BP701** £8.49

KINDLE FIRE HDX EXPLAINED
118 Pages **Order code BP743** £8.99

BOOK ORDER FORM

Full name: ....

Address: ....

....

.... Post code: .... Telephone No: ....

Email: ....

☐ I enclose cheque/PO payable to Practical Electronics for £ ....

☐ Please charge my card £ .... Card expiry date...

Card Number ....

Valid From Date .... Card Security Code .... (The last three digits on or just below the signature strip)

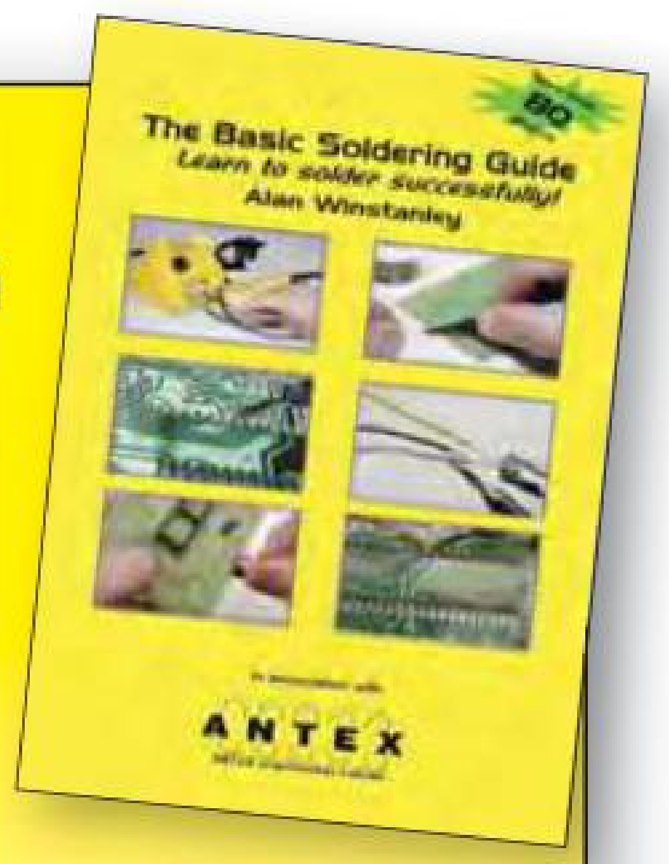
Please send book order codes: ....

....

....

Please continue on separate sheet of paper if necessary

THE BASIC SOLDERING GUIDE



LEARN TO SOLDER SUCCESSFULLY!
ALAN WINSTANLEY

The No.1 resource for learning all the basic aspects of electronics soldering by hand.

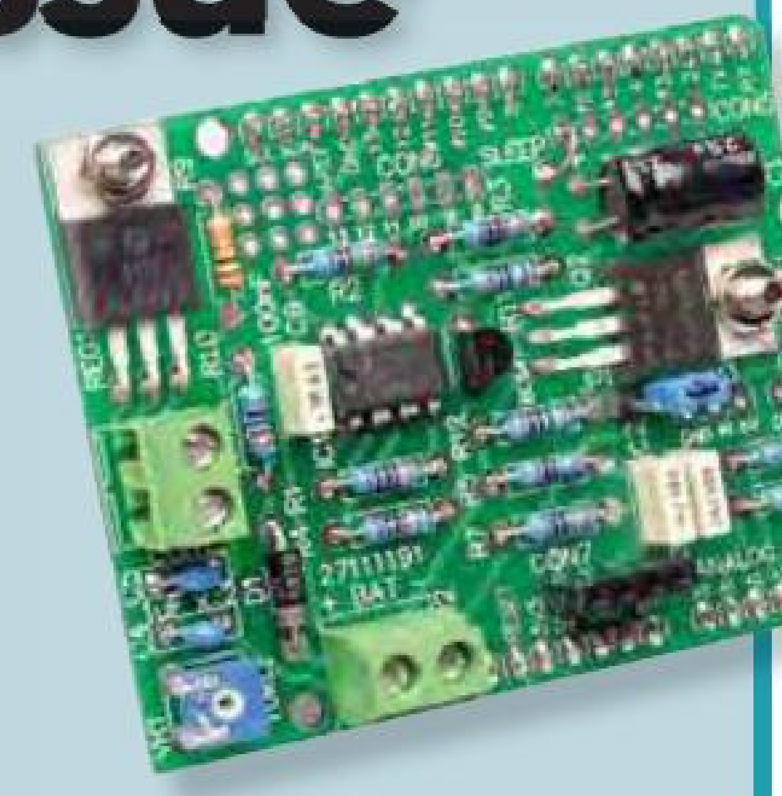
With more than 80 high quality colour photographs, this book explains the correct choice of soldering irons, solder, fluxes and tools. The techniques of how to solder and desolder electronic components are then explained in a clear, friendly and non-technical fashion so you'll be soldering successfully in next to no time! The book also includes sections on reflow soldering and desoldering techniques, potential hazards, useful resources and a very useful troubleshooting guide.

Also ideal for those approaching electronics from other industries, the *Basic Soldering Guide Handbook* is the best resource of its type, and thanks to its excellent colour photography and crystal clear text, the art of soldering can now be learned by everyone!

86 Pages **Order code AW1** £9.99

VISIT OUR WEBSITE FOR MORE BOOKS AND FAST, EASY ONLINE ORDERING: www.electronpublishing.com

Next Month – in the February issue



Remote Monitoring Station

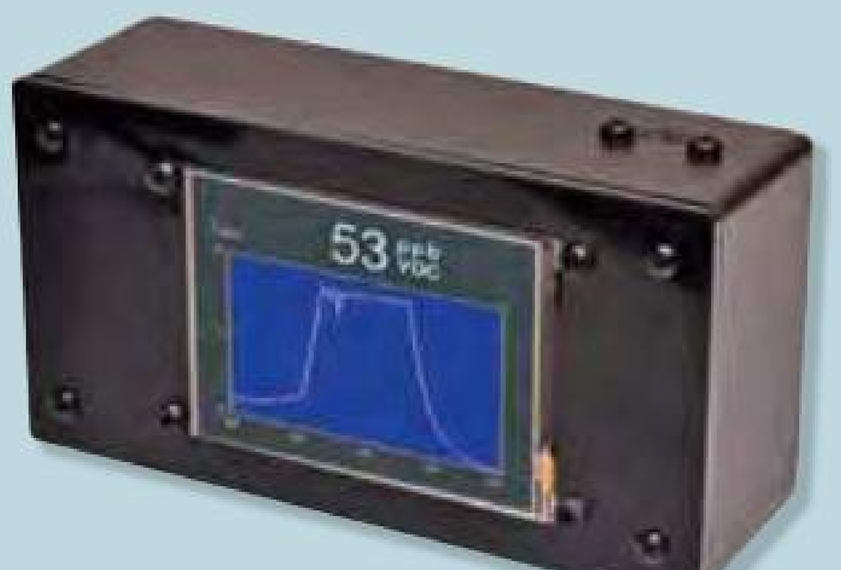
If you have an expensive car, boat, caravan, holiday house, farm… you need to know what's going on when you are away. Is the battery going flat? Is your boat taking on water? All you need to do this is a couple of Arduino shields and a little software. You can even remotely trigger actions, such as switching off misbehaving equipment!

Using Cheap Electronic Modules

Next month, we're looking at an 8-channel USB logic analyser. It is essentially a clone of the original version of the well-known and respected Saleae Logic unit. It's completely compatible with the Saleae design, but you can get this one for as little as £6 – much less than one-twentieth the cost!

Low Distortion DDS Signal Generator

This two-channel audio signal generator produces very low distortion sinewaves, triangle waves, square waves, pulse trains and noise. It has adjustable output frequency, amplitude and phase, plus sweep and pulse modes. It's ideal for testing amplifiers, loudspeakers and all sorts of audio equipment, as well as for general purpose use.



Air Quality Monitor

Measure air quality with a volatile organic compound (VOC) meter. The mighty Micromite BackPack and a cheap module make building one of these dead easy!

PLUS!

All your favourite regular columns from *Audio Out, Cool Beans and Circuit Surgery*, to *Make it with Micromite*, *Practically Speaking* and *Net Work*.

On sale 7 January 2021

Content may be subject to change

WELCOME TO JPG ELECTRONICS

Selling Electronics in Chesterfield for 29 Years

Open Monday to Friday 9am to 5:30pm
And Saturday 9:30am to 5pm

- Aerials, Satellite Dishes & LCD Brackets
- Audio Adaptors, Connectors & Leads
- BT, Broadband, Network & USB Leads
- Computer Memory, Hard Drives & Parts
- DJ Equipment, Lighting & Supplies
- Extensive Electronic Components
  - *ICs, Project Boxes, Relays & Resistors*
- Raspberry Pi & Arduino Products
- Replacement Laptop Power Supplies
- Batteries, Fuses, Glue, Tools & Lots more…

T: 01246 211 202
E: sales@jpgelectronics.com
JPG Electronics, Shaw's Row,
Old Road, Chesterfield, S40 2RB
W: www.jpgelectronics.com

JPG Electronics

Retail & Trade Welcome • Free Parking • Google St View Tour: S40 2RB

NEW subscriptions hotline!

Practical Electronics

We have changed the way we sell and renew subscriptions. We now use 'Select Publisher Services' for all print subscriptions – to start a new subscription or renew an existing one you have three choices:

1. Call our NEW print subscription hotline: 01202 087631, or email: pesubs@selectps.com
2. Visit our shop at: www.electronpublishing.com
3. Send a cheque (payable to: 'Practical Electronics') with your details to:
   Practical Electronics Subscriptions, PO Box 6337,
   Bournemouth BH1 9EH, United Kingdom

Remember, we print the date of the last issue of your current subscription in a box on the address sheet that comes with your copy.

Digital subscribers, please call 01202 880299 or visit: www.electronpublishing.com

Published on approximately the first Thursday of each month by Electron Publishing Limited, 1 Buckingham Road, Brighton, East Sussex BN1 3RA. Printed in England by Acorn Web Offset Ltd., Normanton WF6 1TW. Distributed by Seymour, 86 Newman St., London W1T 3EX. Subscriptions UK: £26.99 (6 months); £49.85 (12 months); £94.99 (2 years). EUROPE: airmail service, £30.99 (6 months); £57.99 (12 months); £109.99 (2 years). REST OF THE WORLD: airmail service, £37.99 (6 months); £70.99 (12 months); £135.99 (2 years). Payments payable to 'Practical Electronics', Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom. Email: pesubs@selectps.com. PRACTICAL ELECTRONICS is sold subject to the following conditions, namely that it shall not, without the written consent of the Publishers first having been given, be lent, resold, hired out or otherwise disposed of by way of Trade at more than the recommended selling price shown on the cover, and that it shall not be lent, resold, hired out or otherwise disposed of in a mutilated condition or in any unauthorised cover by way of Trade or affixed to or as part of any publication or advertising, literary or pictorial matter whatsoever.

Did you know our online shop now sells the current issue of *PE* for £4.99 inc. p&p?

PE
Practical Electronics
The UK's premier electronics and computing maker magazine
PIC n' Mix
PIC18F development board
Make it with Micromite
Counting pulses and rotary encoders
Audio Out
Germanium BJT amplifier
Circuit Surgery
Interference and noise
Toot toot! Complete Arduino DCC Controller
ARDUINO
WIN! Microchip SAM IoT WG Development Board WIN!
NeoPixel gamma correction
Amazing Nutube miniature valve stereo preamplifier
Fascinating XOD/Arduino thermometer
XOD
ARDUINO
Fun LED Christmas Tree offer!
PLUS!
Techno Talk – Lights out for dangerous bulbs?
KickStart – NEW series! – using the 2N7000 MOSFET
Net Work – New electric cars and Android TV
Jan 2021 £4.99
www.electronpublishing.com
@practicalelec
practicalelectronics

You read that right! We now sell the current issue of your favourite electronics magazine for exactly the same price as in the High Street, but we deliver it straight to your door – and for UK addresses we pay the postage. No need to journey into town to queue outside the newsagent. Just go to our website, set up an account in 30 seconds, order your magazine and we'll do the rest.

www.electronpublishing.com

PEAK®
electronic design ltd

15% discount!

Use code *PE15* during checkout

limited time only!

ZEN50 Zener Diode Analyser (inc. LEDs, TVSs etc)

Popular product!

The ***Atlas ZEN*** (model ZEN50) is ideal for testing Zeners, VDRs, transient suppressors, LED strings and many other components.

- **Measure Zener Voltage (from 0.00 up to 50.00V!)**
- **Measure Slope Resistance.**
- **Selectable test current: 2mA, 5mA, 10mA and 15mA.**
- **Very low duty cycle to minimise temperature rise.**
- **Continuous measurements.**
- **Single AAA battery (included) with very long battery life.**
- **Gold plated croc clips included.**
- **Can measure forward voltage of LEDs and LED strings too.**

atlas ZEN
model ZEN50
PEAK electronic design ltd
Vz=27.74V @ 15mA
Slope R=53Ω
Zener diode analyser
Zener Voltage
Slope Resistance
Selectable Test Currents
Vz<50V
on - test
scroll - off
warning: not for in-circuit

~~£46.50~~ **£39.53**
~~£38.75+VAT~~ **with discount!**

LCR45 LCR and Impedance Meter with Auto and Manual modes

Advanced firmware!

The powerful LCR45 meter that not only identifies and measures your passive components (Inductors, Capacitors and Resistors) but also measures complex impedance, magnitude of impedance with phase and admittance too! Auto and Manual test modes allow you to specify the test frequency and component type.

- **Continuous fluid measurements.**
- **Improved measurement resolution: (<0.2µH, <0.2pF).**
- **Test frequencies: DC, 1kHz, 15kHz, 200kHz.**
- **Measure the true impedance of speakers and more.**
- **Great for hobbyists and professionals.**

LCR and impedance meter
PEAK
Inductor 0.31Ω
L=701.1µH 15kHz
LCR45
on - menu
enter - off
warning: avoid charge/voltages
Complex Impedance
Impedance 200kHz
+25.6 +j155.1Ω

~~£91.50~~ **£77.78**
~~£76.25+VAT~~ **with discount!**

DCA55 Semiconductor Analyser - Identify your semi's

Now with backlit display and AAA battery!

Connect any way round to identify the type of component and the pinout! Also measures many parameters including transistor gain, base-emitter voltages, MOSFET thresholds, LED voltages etc. Complete with a comprehensive illustrated user guide. Includes an Alkaline battery so you're ready to go straight away.

- **Transistors (including NPN/PNP, darlingtons, Si & Ge).**
- **Measure hFE, VBE and leakage.**
- **Diodes and LEDs. Measure VF.**
- **MOSFETs. Measure VGS(th).**
- **Gold plated hook probes.**
- **Long battery life.**
- **Free technical support for life.**
- **Comprehensive instruction book.**
- **2 year warranty.**

atlas DCA
model DCA55
PEAK electronic design ltd
AAA Battery Edition
Base-Emitter V
VBE=1.039V
semiconductor component analyser
automatic lead identification
on - test
scroll - off
warning: avoid charge/voltages

~~£52.50~~ **£44.63**
~~£43.75+VAT~~ **with discount!**

ESR70 Equivalent Series Resistance and Capacitance Meter

Now with backlit display and AAA battery!

The new improved ESR70 is great for identifying capacitors (especially electrolytics) that are suffering from high ESR. ESR is a great indicator of capacitor condition, especially in applications demanding high ripple currents such as reservoir capacitors, SMPSUs, motor capacitors etc. Includes Alkaline AAA battery and a very comprehensive user guide. The ESR70 will even perform a controlled discharge of your caps!

atlas ESR+
model ESR70
PEAK electronic design ltd
Cap = 22.28µF
ESR = 0.48Ω
ESR, low resistance and capacitance meter
+ Audible Alerts
with automatic discharge function
on - test
off

~~£84.00~~ **£71.40**
~~£70.00+VAT~~ **with discount!**

It's only possible to show summary specifications here. Please ask if you'd like detailed data. Further information is also available on our website. Product price refunded if you're not happy.

www.peakelec.co.uk
sales@peakelec.co.uk

Atlas House, 2 Kiln Lane, Harpur Hill Business Park
Buxton, Derbyshire, SK17 9JL, UK. Tel: 01298 70012